Iterative Hard Thresholding with Adaptive Regularization: Sparser Solutions Without Sacrificing Runtime

Kyriakos Axiotis [1] **Maxim Sviridenko** [2]

Abstract

We propose a simple modification to the iterative hard thresholding (IHT) algorithm, which recovers asymptotically sparser solutions as a function of the condition number. When aiming to minimize a convex function $f(\boldsymbol{x})$ with condition number κ subject to $\boldsymbol{x}$ being an s-sparse vector, the standard IHT guarantee is a solution with relaxed sparsity $O(s\kappa^2)$, while our proposed algorithm, *regularized IHT*, returns a solution with sparsity $O(s\kappa)$. Our algorithm significantly improves over ARHT (Axiotis & Sviridenko, 2021b) which also finds a solution of sparsity $O(s\kappa)$, as it does not require re-optimization in each iteration (and so is much faster), is deterministic, and does not require knowledge of the optimal solution value $f(\boldsymbol{x}^*)$ or the optimal sparsity level s. Our main technical tool is an *adaptive regularization* framework, in which the algorithm progressively learns the weights of an ℓ_2 regularization term that will allow convergence to sparser solutions. We also apply this framework to low rank optimization, where we achieve a similar improvement of the best known condition number dependence from κ^2 to κ.

1. Introduction

Sparse optimization is the task of optimizing a function f over s-sparse vectors, i.e. those with at most s non-zero entries. Examples of such optimization problems arise in machine learning, with the goal to make models smaller for efficiency, generalization, or interpretability reasons, and compressed sensing, where the goal is to recover an s-sparse signal from a small number of measurements. A closely related problem is *low rank optimization*, where the sparsity constraint is instead placed on the spectrum of the solution (which is a matrix). This problem is central in matrix factorization, recommender systems, robust principal components analysis, among other tasks. More generally, *structured sparsity* constraints have the goal of capturing the special structure of a particular task by restricting the set of solutions to those that are "simple" in an appropriate sense. Examples include group sparsity, tree- and graph-structured sparsity. For more on generalized sparsity measures see e.g. (Schmidt, 2018).

Among the huge number of algorithms that have been developed for the sparse optimization problems, three stand out as the most popular ones:

- The **LASSO** (Tibshirani, 1996), which works by relaxing the ℓ_0 (sparsity) constraint to an ℓ_1 constraint, thus convexifying the problem.
- **Orthogonal matching pursuit** (OMP) (Pati et al., 1993), which works by building the solution greedily in an incremental fashion.
- **Iterative hard thresholding** (IHT) (Blumensath & Davies, 2009), which performs projected gradient descent on the set of sparse solutions.

Among these, IHT is generally the most efficient, since it has essentially no overhead over plain gradient descent, making it the tool of choice for large-scale applications.

1.1. Iterative Hard Thresholding (IHT)

Consider the *sparse convex optimization* problem

$$\min_{\|\boldsymbol{x}\|_0 \le s} f(\boldsymbol{x}) \,, \tag{1}$$

where f is convex and $\|\boldsymbol{x}\|_0$ is the number of non-zero entries in the vector $\boldsymbol{x}$, i.e. the sparsity of $\boldsymbol{x}$. IHT works by repeatedly performing the following iteration

$$\boldsymbol{x}^{t+1} = H_{s'}\left(\boldsymbol{x}^t - \eta \cdot \nabla f(\boldsymbol{x}^t)\right) \,, \tag{2}$$

where $H_{s'}$ is the *hard thresholding operator* that zeroes out all but the top s' entries, for some (potentially relaxed) sparsity level s', and $\eta > 0$ is the step size.

[1]MIT [2]Yahoo! Research. Correspondence to: Kyriakos Axiotis <kaxiotis@mit.edu>, Maxim Sviridenko <sviri@yahooinc.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

As (1) is known to be NP-hard (Natarajan, 1995) and even hard to approximate (Foster et al., 2015), an extra assumption needs to be made for the performance of the algorithm to be theoretically evaluated in a meaningful way. The most common assumption is that the *(restricted) condition number* of f is bounded by κ (or the *restricted isometry property* constant is bounded by δ (Candes, 2008)), but other assumptions have been studied as well, such as incoherence (Donoho & Elad, 2003) and weak supermodularity (Liberty & Sviridenko, 2017). The performance is then measured in terms of the sparsity s' of the returned solution, as well as its error (value of f).

As it is known (Jain et al., 2014), IHT is guaranteed to return an $s' = O(s\kappa^2)$-sparse solution $\boldsymbol{x}$ with $f(\boldsymbol{x}) \leq f(\boldsymbol{x}^*) + \varepsilon$. In fact, as we show in Section E, the κ^2 factor cannot be improved in the analysis. Recently, (Axiotis & Sviridenko, 2021b) presented an algorithm called ARHT, which improves the sparsity to $s' = O(s\kappa)$. However, their algorithm is much less efficient than IHT, for many reasons. So the question emerges:

Is there a sparse convex optimization algorithm that returns $O(s\kappa)$-sparse solutions, but whose runtime efficiency is comparable to IHT?

The main contribution of our work is to show that this goal can be achieved, and done so by a surprisingly simple tweak to IHT.

1.2. Reconciling Sparsity and Efficiency: Regularized IHT

Our main result is the following theorem, which states that running IHT on an *adaptively regularized* objective function returns $O(s\kappa)$-sparse solutions that are ε-optimal in function value, while having no significant runtime overhead over plain gradient descent.

Theorem 1.1 (Regularized IHT)**.** *Let $f \in \mathbb{R}^n \to \mathbb{R}$ be a convex function that is β-smooth and α-strongly convex[1], with condition number $\kappa = \beta/\alpha$, and $\boldsymbol{x}^*$ be an (unknown) s-sparse solution. Then, running Algorithm 1 with $\eta = (2\beta)^{-1}$ and $c = s'/(4T)$ for*

$$T = O\left(\kappa \log \frac{f(\boldsymbol{x}^0) + (\beta/2)\left\|\boldsymbol{x}^0\right\|_2^2 - f(\boldsymbol{x}^*)}{\varepsilon}\right)$$

iterations starting from an arbitrary $s' = O(s\kappa)$-sparse solution $\boldsymbol{x}^0$, the algorithm returns an s'-sparse solution $\boldsymbol{x}^T$ such that $f(\boldsymbol{x}^T) \leq f(\boldsymbol{x}^*) + \varepsilon$. Furthermore, each iteration requires $O(1)$ evaluations of f, ∇f, and $O(n)$ additional time.

[1]The theorem also holds if the smoothness and strong convexity constants are replaced by $(s'+s)$-*restricted* smoothness and strong convexity constants.

To achieve this result, we significantly refine and generalize the *adaptive regularization* technique of (Axiotis & Sviridenko, 2021b). This refined version fixes many of the shortcomings of the original, by (i) not requiring *re-optimization* in every iteration (a relic of OMP-style algorithms), (ii) taking $\widetilde{O}(\kappa)$ instead of $\widetilde{O}(s\kappa)$ iterations, (iii) being *deterministic*, (iv) not requiring knowledge of the optimal function value $f(x^*)$ thus avoiding the overhead of an outer binary search, and (v) being more easily generalizable to other settings, like low rank minimization.

In short, our main idea is to run IHT on a regularized function

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) + (\beta/2)\left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2 ,$$

where $\left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2 = \sum_{i=1}^n w_i x_i^2$ and $\boldsymbol{w}$ are non-negative weights. These weights change dynamically during the algorithm, in a way that depends on the value of $\boldsymbol{x}$. The effect is that now the IHT step will instead be given by

$$\boldsymbol{x}^{t+1} = H_{s'}\left(\left(\mathbf{1} - 0.5\boldsymbol{w}^t\right)\boldsymbol{x}^t - \eta \cdot \nabla f(\boldsymbol{x}^t)\right) ,$$

which is almost the same as (2), except that it has an extra term that biases the solution towards $\mathbf{0}$. Additionally, in each step the weights $\boldsymbol{w}^t$ are updated based on the current solution as

$$w_i^{t+1} = \left(w_i^t \cdot \left(\mathbf{1} - c \cdot \frac{w_i^t (x_i^t)^2}{\left\|\boldsymbol{x}^t\right\|_{w^t,2}^2}\right)\right)_{\geq 1/2}$$

for some parameter $c > 0$, where $(\cdot)_{\geq 1/2}$ denotes zeroing out all the entries that are $< 1/2$ and keeping the others intact.

In Section 3, we will go over the central ideas of our refined adaptive regularization technique, and also explain how it can be extended to deal with more general sparsity measures.

1.3. Beyond Sparsity: Low Rank Optimization

As discussed, our new techniques transfer to the problem of minimizing a convex function under a rank constraint. In particular, we prove the following theorem:

Theorem 1.2 (Adaptive Regularization for Low Rank Optimization)**.** *Let $f \in \mathbb{R}^{m\times n} \to \mathbb{R}$ be a convex function with condition number κ and consider the low rank minimization problem*

$$\min_{\mathrm{rank}(\boldsymbol{A}) \leq r} f(\boldsymbol{A}) . \tag{3}$$

For any error parameter $\varepsilon > 0$, there exists a polynomial time algorithm that returns a matrix $\boldsymbol{A}$ with $\mathrm{rank}(\boldsymbol{A}) \leq O\left(r\left(\kappa + \log \frac{f(\boldsymbol{O}) - f(\boldsymbol{A}^*)}{\varepsilon}\right)\right)$ and $f(\boldsymbol{A}) \leq f(\boldsymbol{A}^*) + \varepsilon$, where $\boldsymbol{O}$ is the all-zero matrix and $\boldsymbol{A}^*$ is any rank-r matrix.

This result can be compared to the Greedy algorithm of (Axiotis & Sviridenko, 2021a), which works by incrementally adding a rank-1 component to the solution and achieves rank $O(r\kappa \log \frac{f(\boldsymbol{O})-f(\boldsymbol{A}^*)}{\varepsilon})$, as well as their Local Search algorithm, which works by simultaneously adding a rank-1 component and removing another, and achieves rank $O(r\kappa^2)$. In contrast, our Theorem 1.2 returns a solution with rank $O\left(r\left(\kappa + \log \frac{f(\boldsymbol{O})-f(\boldsymbol{A}^*)}{\varepsilon}\right)\right)$.

1.4. Related Work

The sparse optimization and compressed sensing literature has a wealth of different algorithms and analyses. Examples include the seminal paper of (Candes, 2008) on recovery with LASSO and followup works (Foucart, 2010), the CoSaMP algorithm (Needell & Tropp, 2009), orthogonal matching pursuit and variants (Natarajan, 1995; Shalev-Shwartz et al., 2010; Jain et al., 2011; Axiotis & Sviridenko, 2021b) iterative hard thresholding (Blumensath & Davies, 2009; Jain et al., 2014), hard thresholding pursuit (Foucart, 2011; Yuan et al., 2016; Shen & Li, 2017a;b), partial hard thresholding (Jain et al., 2017), and message passing algorithms (Donoho et al., 2009). For a survey on compressed sensing, see (Boche et al., 2015; Foucart & Rauhut, 2017).

A family of algorithms that is closely related to IHT are Frank-Wolfe (FW) methods (Frank et al., 1956), which have been used for dealing with generalized sparsity constraints (Jaggi, 2013). The basic version can be viewed as a variant of OMP without re-optimization in each iteration. Block-FW methods are more resemblant of IHT without the projection step, see e.g. (Allen-Zhu et al., 2017) for an application to the low rank minimization problem.

(Liu & Foygel Barber, 2020) presented an interesting connection between hard and soft thresholding algorithms by studying a concavity property of the thresholding operator, and proposed new thresholding operators.

Recently it has been shown (Peste et al., 2021) that IHT can be guaranteed to work for sparse optimization of *non-convex* functions, under appropriate assumptions. In particular, (Peste et al., 2021) studies a stochastic version of IHT for sparse deep learning problems, from both a theoretical and practical standpoint.

2. Background

Notation. We denote $[n] = \{1, 2, \ldots, n\}$. We will use **bold** to refer to vectors or matrices. We denote by $\mathbf{0}$ the all-zero vector, $\mathbf{1}$ the all-one vector, $\boldsymbol{O}$ the all-zero matrix, and by $\boldsymbol{I}$ the identity matrix (with dimensions understood from the context). Additionally, we will denote by $\mathbf{1}_i$ the i-th basis vector, i.e. the vector that is 0 everywhere except at position i.

In order to ease notation and where not ambiguous for two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we denote by $\boldsymbol{xy} \in \mathbb{R}^n$ a vector with elements $(\boldsymbol{xy})_i = x_i y_i$, i.e. the element-wise multiplication of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. In contrast, we denote their inner product by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ or $\boldsymbol{x}^\top \boldsymbol{y}$. Similarly, $\boldsymbol{x}^2 \in \mathbb{R}^n$ will be the element-wise square of vector $\boldsymbol{x}$.

Restrictions and Thresholding. For any vector $\boldsymbol{x} \in \mathbb{R}^n$ and set $S \subseteq [n]$, we denote by $\boldsymbol{x}_S$ the vector that results from $\boldsymbol{x}$ after zeroing out all the entries except those in positions given by indices in S. For any $t \in \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^n$, we denote by $\boldsymbol{x}_{\geq t}$ the vector that results from setting all the entries of $\boldsymbol{x}$ that are less than t to 0. For a function $f \in \mathbb{R}^n \to \mathbb{R}$, its gradient $\nabla f(\boldsymbol{x})$, and a set of indices $S \subseteq [n]$, we denote $\nabla_S f(\boldsymbol{x}) = (\nabla f(\boldsymbol{x}))_S$. We define the *thresholding operator* $H_s(\boldsymbol{x})$ for any vector $\boldsymbol{x}$ as $\boldsymbol{x}_S$, where S are the s entries of $\boldsymbol{x}$ with largest absolute value (breaking ties arbitrarily). We override the thresholding operator $H_r(\boldsymbol{A})$ when the argument is a matrix $\boldsymbol{A}$, defining $H_r(\boldsymbol{A}) = \boldsymbol{U}\operatorname{diag}(H_r(\boldsymbol{\lambda}))\,\boldsymbol{V}^\top$, where $\boldsymbol{U}\operatorname{diag}(\boldsymbol{\lambda})\,\boldsymbol{V}^\top$ is the singular value decomposition of $\boldsymbol{A}$, i.e. $H_r(\boldsymbol{A})$ only keeps the top r singular components of $\boldsymbol{A}$.

Norms and Inner Products. For any $p \in (0, \infty)$ and weight vector $\boldsymbol{w} \geq \mathbf{0}$, we define the weighted ℓ_p norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ as:

$$\|\boldsymbol{x}\|_{p,\boldsymbol{w}} = \left(\sum_i w_i x_i^p\right)^{1/p}.$$

For $p = 0$, we denote $\|\boldsymbol{x}\|_0 = |\{i \mid x_i \neq 0\}|$ to be the *sparsity* of $\boldsymbol{x}$. For $p = \infty$, we denote $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$ to be the maximum absolute value of $\boldsymbol{x}$.

For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we let $\|\boldsymbol{A}\|_2$ be its spectral norm, $\|\boldsymbol{A}\|_F$ be its Frobenius norm, and $\|\boldsymbol{A}\|_*$ be its nuclear norm (i.e. sum of singular values). For any $\boldsymbol{B} \in \mathbb{R}^{m \times n}$, we denote the Frobenius inner product as $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \operatorname{Tr}\left[\boldsymbol{A}^\top \boldsymbol{B}\right]$.

Smoothness, strong convexity, condition number. A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is called *convex* if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ we have $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle$. Furthermore, f is called β-*smooth* for some real number $\beta > 0$ if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ we have $f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + (\beta/2)\|\boldsymbol{y} - \boldsymbol{x}\|_2^2$ and α-*strongly convex* for some real number $\alpha > 0$ if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ we have $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + (\alpha/2)\|\boldsymbol{y} - \boldsymbol{x}\|_2^2$. We call $\kappa := \beta/\alpha$ the *condition number* of f. If f is only β-smooth along s-sparse directions (i.e. only for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ such that $\|\boldsymbol{y} - \boldsymbol{x}\|_0 \leq s$), then we call f β-smooth *at sparsity level* s and denote the smallest such β by β_s and call it the *restricted smoothness constant* (at sparsity level s). We analogously define the *restricted strong convexity constant* α_s, as well as the s-*restricted condition number* $\kappa_s := \beta_s/\alpha_s$.

Projections. Given a subspace $\mathcal{V}$, we will denote the orthogonal projection onto $\mathcal{V}$ as $\boldsymbol{\Pi}_{\mathcal{V}}$. In particular, for any matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ we denote by $\mathrm{im}(\boldsymbol{A}) = \{\boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^n\}$ the *image* of $\boldsymbol{A}$ and by $\ker(\boldsymbol{A}) = \{\boldsymbol{x} \mid \boldsymbol{A}^\top \boldsymbol{x} = \boldsymbol{0}\}$ the *kernel* of $\boldsymbol{A}$. Therefore, $\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{A})} = \boldsymbol{A}\left(\boldsymbol{A}^\top \boldsymbol{A}\right)^+ \boldsymbol{A}^\top$ is the orthogonal projection onto the image of $\boldsymbol{A}$ and $\boldsymbol{\Pi}_{\ker(\boldsymbol{A}^\top)} = \boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{A})}$ the orthogonal projection onto the kernel of $\boldsymbol{A}^\top$, where $(\cdot)^+$ denotes the matrix pseudoinverse.

3. The Adaptive Regularization Method

Consider the sparse optimization problem

$$\min_{\|\boldsymbol{x}\|_0 \le s} f(\boldsymbol{x}) \tag{4}$$

on a convex function f with condition number at most κ, and an optimal solution $\boldsymbol{x}^*$ that is supported on the set of indices $S^* \subseteq [n]$.

The main hurdle towards solving this problem is that it is NP hard. Therefore, it is common to relax it by a factor depending on κ. In fact, IHT requires relaxing the sparsity constraint by a factor of $O(\kappa^2)$ (i.e. $\|\boldsymbol{x}\|_0 \le O(s\kappa^2)$), in order to return a near-optimal solution. Also, the κ^2 factor is tight for IHT (see Appendix E).

Remark We state all our results in terms of the condition number κ, even though the statements can be strengthened to depend on the *restricted* condition number $\kappa_{s'+s}$, specifically the condition number restricted on $(s'+s)$-sparse directions. We state our results in this weaker form for clarity of presentation.

3.1. Regularized IHT

Perhaps surprisingly, there is a way to *regularize* the objective by a weighted ℓ_2 norm so that running IHT on the new objective will only require relaxing the sparsity by $O(\kappa)$:

$$\min_{\|\boldsymbol{x}\|_0 \le s} f(\boldsymbol{x}) + (\beta/2)\,\|\boldsymbol{x}\|_{\boldsymbol{w},2}^2 \,. \tag{5}$$

One way to do this is by setting the weights $\boldsymbol{w}$ to be 1 everywhere except in the indices from S^*, where it is set to 0. An inquisitive reader will protest that this is not a very useful statement, since it requires knowledge of S^*, which was our goal to begin with. In fact, we could just as easily have used the regularizer $(\beta/2)\,\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2$, thus penalizing everything that is far from the optimum!

3.2. Learning Weights

Our main contribution is to show that the optimal weights $\boldsymbol{w}$ can in fact be *learned* in the duration of the algorithm[2]. More precisely, consider running IHT starting from the setting of $\boldsymbol{w} = \boldsymbol{1}$. The regularized objective (5) is now $O(1)$-conditioned, which is great news. On the other hand, (5) is not what we set out to minimize. In other words, even though this approach might work great for minimizing (5), it might (and generally will) fail to achieve sufficient decrease in (4)—one could view this as the algorithm getting trapped in a local minimum.

Our main technical tool is to characterize these local minima, by showing that they can only manifest themselves if the current solution $\boldsymbol{x}$ satisfies the following condition:

$$\|\boldsymbol{x}_{S^*}\|_{\boldsymbol{w},2}^2 \ge \Omega(\kappa^{-1})\,\|\boldsymbol{x}\|_{\boldsymbol{w},2}^2 \,. \tag{6}$$

In words, this means that a significant fraction of the mass of the current solution lies in the support S^* of the optimal solution. Interestingly, this gives us enough information based on which to update the regularization weights $\boldsymbol{w}$ in a way that the sum of weights in S^* drops fast enough compared to the total sum of weights. This implies that the vector $\boldsymbol{w}$ moves in a direction that correlates with the direction of the optimal weight vector.

These are the core ideas needed to bring the sparsity overhead of IHT from $O(\kappa^2)$ down to $O(\kappa)$.

3.3. Beyond Sparsity: Learning Subspaces

One can summarize the approach of the previous section in the following more general way: If we know that the optimal solution $\boldsymbol{x}^*$ lies in a particular low-dimensional subspace (in our case this was the span of $\boldsymbol{1}_i$ for all $i \in S^*$), then we can define a regularization term that penalizes all the solutions based on their distance to that subspace. Of course, this subspace is unknown to us, but we can try to adaptively modify the regularization term every time the algorithm gets stuck, just as we did in the previous section.

More concretely, given a collection $\mathcal{A}$ of unit vectors from $\mathbb{R}^n$ (commonly called *atoms*), we define the following problem:

$$\min_{\mathrm{rank}_{\mathcal{A}}(\boldsymbol{x}) \le r} f(\boldsymbol{x})\,, \tag{7}$$

where $\mathrm{rank}_{\mathcal{A}}(\boldsymbol{x})$ is the smallest number of vectors from $\mathcal{A}$ such that $\boldsymbol{x}$ can be written as their linear combination.

[2]The idea of adaptively learning regularization weights looks on the surface similar to adaptive gradient algorithms such as AdaGrad (Duchi et al., 2011). An important difference is that these algorithms regularize the function around the current solution, while we regularize it around the origin. Still, this is a potentially intriguing connection that deserves to be investigated further.

We can pick $\mathcal{A} = \{\mathbf{1}_1, \mathbf{1}_2, \ldots, \mathbf{1}_n\}$ to obtain the sparse optimization problem, $\mathcal{A} = \{\mathrm{vec}(\boldsymbol{u}\boldsymbol{v}^\top) \mid \|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1\}$ for the low rank minimization problem, and other choices of $\mathcal{A}$ can capture more sophisticated problem constraints such as graph structure. Defining an IHT variant for these more general settings is usually straightforward, although the analysis for even obtaining a rank overhead of $O(\kappa^2)$ does not trivially follow and depends on the structure of $\mathcal{A}$.

So, how would a regularizer look in this more general setting? Given our above discussion, it is fairly simple to deduce it. Consider a decomposition of $\boldsymbol{x}$ as the sum of rank-1 components from $\mathcal{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{|\mathcal{A}|}\}$:

$$\boldsymbol{x} = \sum_{i \in S} \boldsymbol{a}_i \,,$$

where $\mathrm{rank}_{\mathcal{A}}(\boldsymbol{a}_i) = 1$, and let $L^* = \mathrm{span}(\{\boldsymbol{a}_i \mid i \in S^*\})$ be a low-dimensional subspace that contains the optimal solution and $L^*_\perp$ is its complement. We can then define the regularizer

$$\Phi^*(\boldsymbol{x}) = (\beta/2) \sum_{i \in S} \left\| \boldsymbol{\Pi}_{L^*_\perp} \boldsymbol{a}_i \right\|_2^2 \,,$$

where $\boldsymbol{\Pi}_{L^*_\perp}$ is the orthogonal projection onto the subspace perpendicular to L^*—in other words $\left\| \boldsymbol{\Pi}_{L^*_\perp} \boldsymbol{a}_i \right\|_2$ is the ℓ_2 distance from $\boldsymbol{a}_i$ to L^*. An equivalent but slightly more concise way is to write:

$$\Phi^*(\boldsymbol{x}) = (\beta/2) \left\langle \boldsymbol{\Pi}_{L^*_\perp}, \sum_{i \in S} \boldsymbol{a}_i \boldsymbol{a}_i^\top \right\rangle \,.$$

Then, we can replace the unknown projection matrix $\boldsymbol{\Pi}_{L^*_\perp}$ by a weight matrix $\boldsymbol{W}$ initialized at $\boldsymbol{I}$, and proceed by adaptively modifying $\boldsymbol{W}$ as we did in the previous section.

It should be noted that the full analysis of this framework is not automatic for general $\mathcal{A}$, and there are several technical challenges that arise depending on the choice of $\mathcal{A}$. In particular, it does not directly apply to the low rank minimization case, and we end up using a different choice of regularizer. However, the discussion in this section should serve as a basic framework for improving the IHT analysis in more general settings, as in particular it did to motivate the low rank optimization analysis that we will present in Section 5.

4. Sparse Optimization Using Regularized IHT

The main result of this section is an efficient algorithm for sparse optimization of convex functions that, even though is a slight modification of IHT, improves the sparsity by an $O(\kappa)$ factor, where κ is the condition number. The regularized IHT algorithm is presented in Algorithm 1 and its analysis is in Theorem 1.1, whose proof can be found in Appendix B.

Algorithm 1 Regularized IHT

$\boldsymbol{x}^0$: initial s'-sparse solution
$\boldsymbol{w}^0 = \mathbf{1}$: initial regularization weights
η: step size, T: #iterations
c: weight step size
for $t = 0 \ldots T-1$ **do**
  $\boldsymbol{x}^{t+1} = H_{s'}\left((\mathbf{1} - 0.5\boldsymbol{w}^t)\boldsymbol{x}^t - \eta \cdot \nabla f(\boldsymbol{x}^t)\right)$
  $\boldsymbol{w}^{t+1} = \left(\boldsymbol{w}^t - c \cdot (\boldsymbol{w}^t \boldsymbol{x}^t)^2 / \left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^t,2}\right)_{\geq 1/2}$
  if $f(\boldsymbol{x}^{t+1}) + (4\eta)^{-1}\left\|\boldsymbol{x}^{t+1}\right\|^2_{\boldsymbol{w}^{t+1},2} > f(\boldsymbol{x}^t) + (4\eta)^{-1}\left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^{t+1},2}$ **then**
    $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t$ {In practice there is no need to perform this step.}
  end if
end for

Theorem 1.1 (Regularized IHT)**.** Let $f \in \mathbb{R}^n \to \mathbb{R}$ be a convex function that is β-smooth and α-strongly convex, with condition number $\kappa = \beta/\alpha$, and $\boldsymbol{x}^*$ be an (unknown) s-sparse solution. Then, running Algorithm 1 with $\eta = (2\beta)^{-1}$ and $c = s'/(4T)$ for

$$T = O\left(\kappa \log \frac{f(\boldsymbol{x}^0) + (\beta/2)\left\|\boldsymbol{x}^0\right\|_2^2 - f(\boldsymbol{x}^*)}{\varepsilon}\right)$$

iterations starting from an arbitrary $s' = O(s\kappa)$-sparse solution $\boldsymbol{x}^0$, the algorithm returns an s'-sparse solution $\boldsymbol{x}^T$ such that $f(\boldsymbol{x}^T) \leq f(\boldsymbol{x}^*) + \varepsilon$. Furthermore, each iteration requires $O(1)$ evaluations of f, ∇f, and $O(n)$ additional time.

The main ingredient for proving Theorem 1.1 is Lemma 4.1, which states that each step of the algorithm either makes substantial (multiplicative) progress in an appropriately regularized function $f(\boldsymbol{x}) + (\beta/2)\left\|\boldsymbol{x}\right\|^2_{w,2}$, or a significant fraction of the mass of $\boldsymbol{x}^2$ lies in S^*, which is the support of the target solution. This latter condition allows us to adapt the weights $\boldsymbol{w}$ in order to obtain a new regularization function that penalizes the target solution less. The proof of the lemma can be found in Appendix C.

Lemma 4.1 (Regularized IHT step progress)**.** *Let $f \in \mathbb{R}^n \to \mathbb{R}$ be a convex function that is β-smooth and α-strongly convex, $\kappa = \beta/\alpha$ be its condition number, and $\boldsymbol{x}^*$ be any s-sparse solution.*

Given any s'-sparse solution $\boldsymbol{x} \in \mathbb{R}^n$ where

$$s' \geq (128\kappa + 2)s$$

and a weight vector $\boldsymbol{w} \in (\{0\} \cup [1/2, 1])^n$ such that $\left\|\boldsymbol{w}\right\|_1 \geq n - s'/2$, we make the following update:

$$\boldsymbol{x}' = H_{s'}\left((\mathbf{1} - 0.5\boldsymbol{w})\boldsymbol{x} - (2\beta)^{-1}\nabla f(\boldsymbol{x})\right) \,.$$

Then, at least one of the following two conditions holds:

- *Updating $\boldsymbol{x}$ makes regularized progress:*

$$g(\boldsymbol{x}') \leq g(\boldsymbol{x}) - (16\kappa)^{-1}(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) ,$$

where

$$g(\boldsymbol{x}) := f(\boldsymbol{x}) + (\beta/2)\left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2$$

is the ℓ_2-regularized version of f with weights given by $\boldsymbol{w}$. Note: The regularized progress statement is true as long as $\boldsymbol{x}$ is suboptimal, i.e. $g(\boldsymbol{x}) > f(\boldsymbol{x}^*)$. Otherwise, we just have $g(\boldsymbol{x}') \leq g(\boldsymbol{x})$.

- *$\boldsymbol{x}$ is significantly correlated to the optimal support $S^* := \operatorname{supp}(\boldsymbol{x}^*)$:*

$$\left\|\boldsymbol{x}_{S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq (4\kappa + 6)^{-1}\left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2 ,$$

and the regularization term restricted to S^* is non-negligible:

$$(\beta/2)\left\|\boldsymbol{x}_{S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq (8\kappa + 8)^{-1}\left(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)\right) .$$

Comparison to ARHT. The ARHT algorithm of (Axiotis & Sviridenko, 2021b) is also able to achieve a sparsity bound of $O(s\kappa)$. However, their algorithm is not practically desirable for a variety of reasons.

- First of all, it follows the OMP (more accurately, OMP with Removals) paradigm, which makes local changes to the support of the solution by inserting or removing a single element of the support, and then *fully re-optimizing* the function on its restriction to this support. Even though the support will generally be very small compared to the ambient dimension n, this is still a significant runtime overhead. In contrast, regularized IHT does not require re-optimization.

  Additionally, the fact that in the ARHT only one new element is added at a time leads to an iteration count that scales with $s\kappa$, instead of the κ of regularized IHT. This is a significant speedup, since both algorithms have to evaluate the gradient in each iteration. Therefore, regularized IHT will require $O(s)$ times fewer gradient evaluations.

- When faced with the non-progress condition, in which the regularized function value does not decrease sufficiently, ARHT moves by selecting a random index i with probability proportional to x_i^2, and proceeds to *unregularize* this element, i.e. remove it from the sum of regularization terms. Instead, our algorithm is completely deterministic. This is achieved by allowing a *weighted* regularization term, and gradually reducing the regularization weights instead of dropping terms.

- ARHT requires knowledge of the optimal function value $f(\boldsymbol{x}^*)$. The reason is that in each iteration they need to gauge whether enough progress was made in reducing the value of the regularized function g, compared to how far it is from the optimal function value. If so, they would perform the unregularization step. In contrast, our analysis does not require these two cases (updates to $\boldsymbol{x}$ or $\boldsymbol{w}$) to be exclusive, and in fact simultaneously updates both, regardless of how much progress was made in g. Thus, our algorithm avoids the expensive overhead of an outer binary search over the optimal value $f(\boldsymbol{x}^*)$.

For all these reasons, as well as its striking simplicity, we believe that regularized IHT can prove to be a useful practical sparse optimization tool.

5. Low Rank Optimization Using Regularized Local Search

In this section we present a regularized local search algorithm for low rank optimization of convex functions, that returns an ε-optimal solution with rank $O\left(r\left(\kappa + \log\frac{f(\boldsymbol{O})-f(\boldsymbol{A}^*)}{\varepsilon}\right)\right)$, where r is the target rank. The algorithm is based on the Local Search algorithm of (Axiotis & Sviridenko, 2021a), but also uses adaptive regularization, which leads to a lot new technical hurdles that are addressed in the analysis. This is presented in Theorem 1.2 and proved in Appendix D.

Theorem 1.2 (Adaptive Regularization for Low Rank Optimization)**.** Let $f \in \mathbb{R}^{m\times n} \to \mathbb{R}$ be a convex function with condition number κ and consider the low rank minimization problem

$$\min_{\operatorname{rank}(\boldsymbol{A})\leq r} f(\boldsymbol{A}) . \tag{8}$$

For any error parameter $\varepsilon > 0$, there exists a polynomial time algorithm that returns a matrix $\boldsymbol{A}$ with $\operatorname{rank}(\boldsymbol{A}) \leq O\left(r\left(\kappa + \log\frac{f(\boldsymbol{O})-f(\boldsymbol{A}^*)}{\varepsilon}\right)\right)$ and $f(\boldsymbol{A}) \leq f(\boldsymbol{A}^*) + \varepsilon$, where $\boldsymbol{O}$ is the all-zero matrix and $\boldsymbol{A}^*$ is any rank-r matrix.

Discussion about ε dependence. Some of the technical issues in the rank case have to do with operator *non-commutativity* and thus pose no issue in the sparsity case. In particular, the extra $\log\frac{f(\boldsymbol{O})-f(\boldsymbol{A}^*)}{\varepsilon}$ dependence in the rank comes exactly because of these issues. However, we think that it should be possible to completely remove this dependence in the future by a more careful analysis.

Discussion about computational efficiency. We note that the goal of this section is to show an improved rank bound, and not to argue about the computational efficiency

of such an algorithm. It might be possible to derive an efficient algorithm by transforming the proof in Theorem 1.2 into a proof for a matrix IHT algorithm, which might be significantly more efficient, as it will not require solving linear systems in each iteration. Still, there are a lot of remaining issues to be tackled, as currently the algorithm requires computing multiple singular value decompositions and orthogonal projections in each iteration. Therefore working on a computationally efficient algorithm that can guarantee a rank of $O(r\kappa)$ is a very interesting direction for future research.

Matrix regularizer Getting back into the main ingredients of Theorem 1.2, we describe the choice of our regularizer. As we are working over general rectangular matrices, we use *two* regularizers, one for the left singular vectors and one for the right singular vectors of $\boldsymbol{A}$. Concretely, given two weight matrices $\boldsymbol{Y}$, $\boldsymbol{W}$ such that $\boldsymbol{O} \preceq \boldsymbol{Y} \preceq \boldsymbol{I}$, $\boldsymbol{O} \preceq \boldsymbol{W} \preceq \boldsymbol{I}$, we define

$$\Phi(\boldsymbol{A}) = (\beta/4)\left(\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle + \langle \boldsymbol{Y}, \boldsymbol{A}^\top\boldsymbol{A}\rangle\right) ,$$

where β is a bound on the smoothness of f. The gradient of the regularized function is

$$\nabla g(\boldsymbol{A}) = \nabla f(\boldsymbol{A}) + (\beta/2)\left(\boldsymbol{W}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{Y}\right) ,$$

and the new solution $\overline{\boldsymbol{A}}$ is defined as

$$\overline{\boldsymbol{A}} = H_{s'-1}\left(\boldsymbol{A}\right) - \eta H_1\left(\nabla g(\boldsymbol{A})\right) ,$$

where we remind that the thresholding operator $H_r : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ that is used in the algorithm returns the top r components of the singular value decomposition of a matrix, i.e. given $\boldsymbol{M} = \sum_{i=1}^{k} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$, where $\boldsymbol{\lambda}_1 \geq \cdots \geq \boldsymbol{\lambda}_k$ are the singular values and $r \leq k$, $H_r(\boldsymbol{M}) = \sum_{i=1}^{r} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$. In other words, we drop the bottom rank-1 component of $\boldsymbol{A}$ and add the top rank-1 component of the gradient.

After taking a step, we re-optimize over matrices with the current left and right singular space, also known as performing a fully corrective step, as in (Shalev-Shwartz et al., 2011; Axiotis & Sviridenko, 2021a). To do this, we first compute the SVD $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ of $\overline{\boldsymbol{A}}$ and then solve the optimization problem $\min_{\boldsymbol{A}=\boldsymbol{U}\boldsymbol{X}\boldsymbol{V}^\top} g^t(\boldsymbol{A})$. For simplicity we assume that this optimization problem can be solved exactly, but the analysis can be modified to account for the case when we have an approximate solution and we are only given a bound on the norm of the gradient (projected onto the relevant subspace), i.e. $\left\|\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\nabla g^t(\boldsymbol{A})\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\right\|_F$.

Whenever there is not enough progress, we make the following updates on the weight matrices $\boldsymbol{W}$ and $\boldsymbol{Y}$:

$$\boldsymbol{W}' = \boldsymbol{W} - \boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}/\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle$$
$$\boldsymbol{Y}' = \boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Y}/\langle \boldsymbol{Y}, \boldsymbol{A}^\top\boldsymbol{A}\rangle .$$

The full algorithm description is in Algorithm 2, Appendix D. In the algorithm description we assume that $f(\boldsymbol{A}^*)$ is known. This assumption can be removed by performing binary search over this value, as in (Axiotis & Sviridenko, 2021b).

6. Experiments

Introduction. In this section we present numerical experiments in order to compare the performance of IHT and regularized IHT (Algorithm 1) in training sparse linear models. In particular, we will look at the tasks of linear regression and logistic regression using both real and synthetic data. In the former, we are given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, where each row represents an example and each column a feature, and a vector $\boldsymbol{b} \in \mathbb{R}^m$ that represents the ground truth outputs, and our objective is to minimize the ℓ_2 loss

$$(1/2)\left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\right\|_2^2 .$$

In logistic regression, $\boldsymbol{b}$ has binary instead of real entries, and our objective is to minimize the logistic loss

$$\underbrace{-\sum_{i=1}^{m}\left(b_i \log\sigma(\boldsymbol{A}\boldsymbol{x})_i + (1-b_i)\log(1-\sigma(\boldsymbol{A}\boldsymbol{x})_i)\right)}_{l(\boldsymbol{x})} ,$$

where $\sigma(z) = \left(1+e^{-z}\right)^{-1}$ is the sigmoid function. As is common, we look at the *regularized* logistic regression objective:

$$l(\boldsymbol{x}) + (\rho/2)\left\|\boldsymbol{x}\right\|_2^2 ,$$

for some $\rho > 0$. For our experiments we use $\rho = 0.1$.

Preprocessing and choice of parameters. The only preprocessing we perform is to center the columns of $\boldsymbol{A}$, i.e. we subtract the mean of each column from each entry of the column, and then scale the columns to unit ℓ_2 norm. This ensures that for any sparsity parameter $s' \in [n]$, the function f is s'-smooth when restricted to s'-sparse directions, or in other words the s'-restricted smoothness constant of f is at most s'. Thus we set our smoothness estimate to $\beta := s'$. Our smoothness estimate β influences the (regularized) IHT algorithm in two ways. First, as the step size of the algorithm is given by $1/\beta$, a value of β that is too large can slow down the algorithm, or even get it stuck to a local minimum. Second, the strength of the regularization term in regularized IHT should be close to the $(s+s')$-restricted smoothness constant, as shown in the analysis of Theorem 1.1.

Even though having a perfectly accurate estimate of the smoothness constant is not necessary, a more accurate estimate improves the performance of the algorithm. In fact, the estimate $1/s'$ for the step size is generally too conservative. When used in practice, one should either tune this parameter or use a variable/adaptive step size to achieve the best results.

For the weight step size of regularized IHT, we set the weight step size to $c = s'/T$, but we also experiment with how changing c affects the performance of the algorithm. The downside of this setting is that it requires knowing the number of iterations a priori. However, in practice one could tune c and then run the algorithm for $O(s'/c)$ iterations. Note that ideally, based on the theoretical analysis, T would be proportional to the restricted condition number of f, however this quantity is hard to compute in general. Another idea to avoid this in practice could be to let c be a variable step size.

Implementation. Both the IHT and regularized IHT algorithms are incredibly simple, and can be described in a few lines of python code, as can be seen in Figure 2, Appendix A. Note that in comparison to Algorithm 1 we do not perform the conditional assignment. All the experiments were run on a single 2.6GHz Intel Core i7 core of a 2019 MacBook Pro with 16GB DDR4 RAM using Python 3.9.10.

6.1. Real data

We first experiment with real data, specifically the *year* regression dataset from UCI (Dua & Graff, 2017) and the *rcv1* binary classification dataset (Lewis et al., 2004), which have been previously used in the literature. Performance on other datasets was similar. In Figure 1 **(a)**-**(b)** we have a comparison between the error of the solution returned by IHT and regularized IHT for a fixed sparsity level. Specifically, if we let $\boldsymbol{x}^{**}$ be the (dense) global minimizer of f, we plot the logarithm of the (normalized) *excess loss* $(f(\boldsymbol{x}) - f(\boldsymbol{x}^{**}))/f(\mathbf{0})$ against the number of iterations. Note that $f(\boldsymbol{x}^{**})$ will typically be considerably lower than the loss of the sparse optimum $f(\boldsymbol{x}^{*})$. In order to make a fair comparison, for each algorithm we pick the best fixed step size of the form $2^i/s$ for integer $i \geq 0$, where s is the fixed sparsity level. The best step sizes of IHT and regularized IHT end up being $2/s, 4/s$ respectively for the linear regression example, and $8/s, 16/s$ respectively for the logistic regression example.

We notice that initially regularized IHT has a much higher error than IHT, but after some iterations it is lower than IHT. This phenomenon is to be expected, because the algorithm runs on a regularized function, and so tries to keep not just $f(\boldsymbol{x})$ but also $\|\boldsymbol{x}\|_2^2$ small. After some iterations, when the algorithm has learnt regularization weights that are closer to the optimal ones, it converges to sparser solutions than IHT (equivalently, lower error solutions with the same sparsity, which is what is shown in the plot).

In Figure 1 **(c)** we compare IHT and regularized IHT for different sparsity levels on the year dataset. If e_1 and e_2 are the excess errors of IHT and regularized IHT respectively, we plot e_2/e_1, which is the relative excess error of regularized IHT with respect to that of IHT. We notice a reduction of up to 40% on the excess error. In Figure 1 **(d)** we examine the effect of the choice of the weight step size c. We conclude that c can give a tradeoff between runtime and accuracy, as setting it to a large value will lead to faster weight decay and thus resemble IHT, while a small value of c will lead to slow weight decrease, which will lead to more iterations but also potentially recover an improved solution. Here we can see an interesting tradeoff between the number of iterations and the error of the solution that is eventually returned. In particular, the *larger* c is, the *faster* the degradation of regularization weights. Thus, for $c \to \infty$, the algorithm tends to be the same as IHT. On the other hand, with *smaller* values of c, one can get an improved error rate, but at the cost of a larger number of iterations. This is because the regularization weights decrease slowly, and so in the early iterations of the algorithm the regularization term will account for a significant fraction of the objective function value.

6.2. Synthetic data

We now turn to synthetically generated linear regression instances. The first result presented in Figure 1 **(e)** is the hard IHT instance that we derived in our lower bound in Appendix E. This experiment shows that there exist examples where, with bad initialization, IHT cannot decrease the objective at all (i.e. is stuck at a local minimum), while regularized IHT with the same initialization manages to reduce the loss by more than 70%.

The second result is a result in the well known setting of sparse signal recovery from linear measurements. We generate a matrix $\boldsymbol{A}$ with entries that are sampled i.i.d. from the standard normal distribution, an s-sparse signal $\boldsymbol{x}$ again with entries sampled i.i.d. from the standard normal distribution, and an observed vector $\boldsymbol{b} := \boldsymbol{A}\boldsymbol{x}$. The goal is to recover $\boldsymbol{x}$ by minimizing the objective

$$f(\boldsymbol{x}) = (1/2)\,\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \,.$$

In Figure 1 **(f)**, we plot the normalized value of this objective, after running both IHT and regularized IHT for the same number of iterations. Here we pick the best step size per instance, starting from $\eta = 1/s$ and increasing in multiples of 1.2. Also, for each fixed value of s and algorithm, we run the experiments 20 times in order to account for the variance. The results show a superiority in the performance of regularized IHT for the sparse signal recovery task.

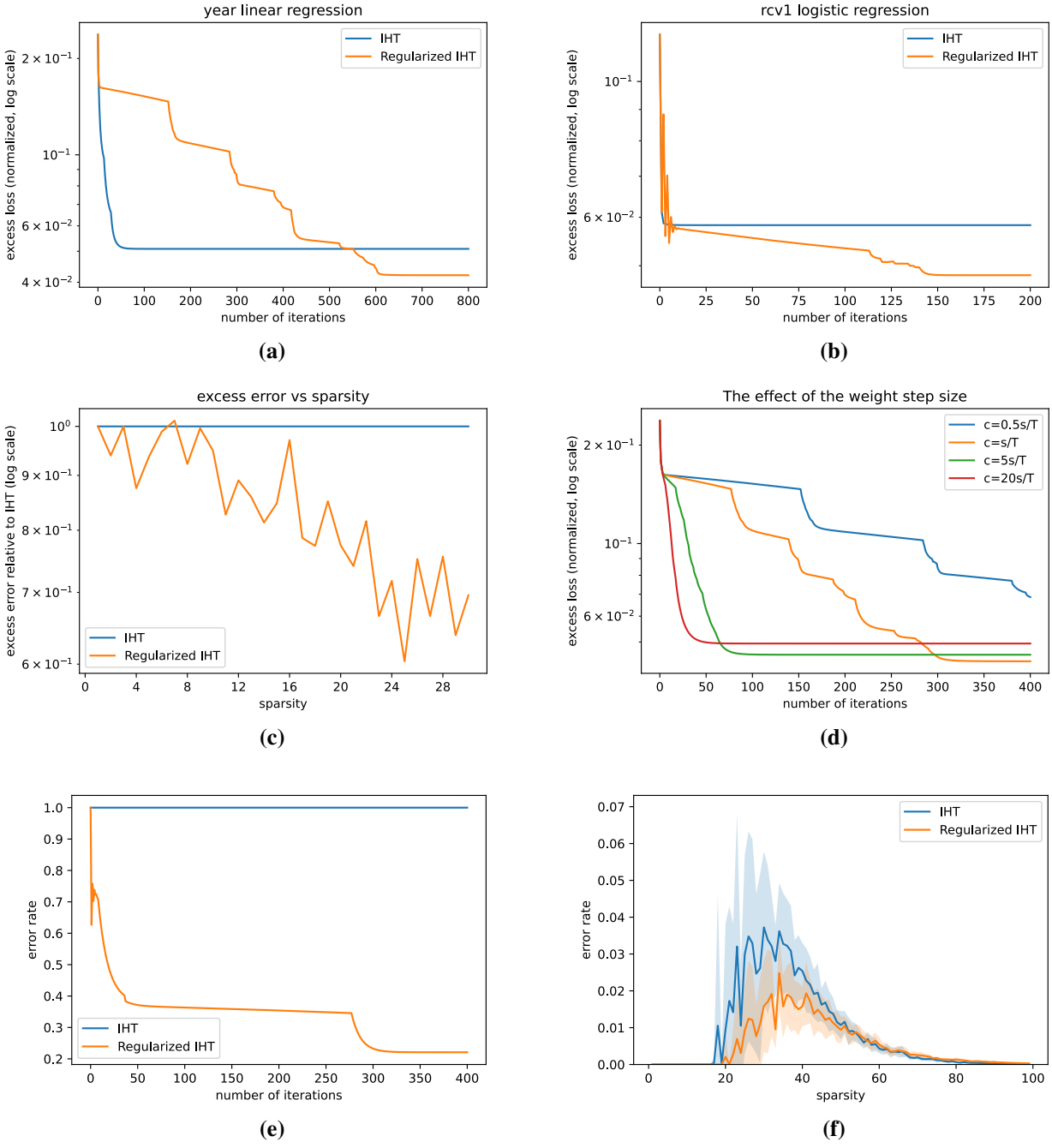

Figure 1. **(a)-(b)**: IHT vs Regularized IHT performance on the year and rcv1 datasets with fixed sparsity levels $s = 11$ and $s = 10$ respectively. On the x axis we have number of iterations and on the y axis we have the normalized excess loss (compared to the dense global optimum), in log scale. The excess loss of regularized IHT is less than that of IHT, specifically 17.3% and 17.2% respectively less in the two experiments. **(c):** Excess error of regularized IHT relative to IHT in the *year* dataset, where sparsity values range from 1 to 30. Both algorithms are run for $T = 800$ iterations. **(d):** Error rate vs number of iterations of regularized IHT on the *year* dataset with fixed sparsity $s = 11$ and step size $\eta = 4/s$, using different values for the weight step size c. **(e):** A demonstration of a 80% decrease in loss by using regularized IHT instead of IHT on the hard instance for IHT presented in Section E. We have generated the data with a condition number of $\kappa = 20$, and a planted sparse solution with sparsity $s = 2$. The dimension is $n = 842$. It can be observed that, for the given initialization vector, IHT never makes any progress on decreasing the error. In contrast, regularized IHT is able to decrease it by almost a factor of 5. **(f):** Sparse signal recovery, where $\boldsymbol{A}$ is an 100×800 measurement matrix, the sparsity level ranges from 1 to 100, and each algorithm is run for 240 iterations. Bands of 1 standard error are shown, after running each data point 20 times independently.

References

Allen-Zhu, Z., Hazan, E., Hu, W., and Li, Y. Linear convergence of a frank-wolfe type algorithm over trace-norm balls. *arXiv preprint arXiv:1708.02105*, 2017.

Axiotis, K. and Sviridenko, M. Local search algorithms for rank-constrained convex optimization. In *International Conference on Learning Representations*, 2021a.

Axiotis, K. and Sviridenko, M. Sparse convex optimization via adaptively regularized hard thresholding. *Journal of Machine Learning Research*, 22:1–47, 2021b.

Blumensath, T. and Davies, M. E. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Boche, H., Calderbank, R., Kutyniok, G., and Vybíral, J. A survey of compressed sensing. In *Compressed sensing and its applications*, pp. 1–39. Springer, 2015.

Candes, E. J. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008.

Donoho, D. L. and Elad, M. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

Donoho, D. L., Maleki, A., and Montanari, A. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Foster, D., Karloff, H., and Thaler, J. Variable selection is hard. In *Conference on Learning Theory*, pp. 696–709, 2015.

Foucart, S. A note on guaranteed sparse recovery via l1-minimization. *Applied and Computational Harmonic Analysis*, 29(1):97–103, 2010.

Foucart, S. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

Foucart, S. and Rauhut, H. A mathematical introduction to compressive sensing. *Bull. Am. Math*, 54:151–165, 2017.

Frank, M., Wolfe, P., et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2): 95–110, 1956.

Horn, R. A. and Johnson, C. Topics in matrix analysis, 1991.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pp. 427–435. PMLR, 2013.

Jain, P., Tewari, A., and Dhillon, I. S. Orthogonal matching pursuit with replacement. In *Advances in neural information processing systems*, pp. 1215–1223, 2011.

Jain, P., Tewari, A., and Kar, P. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pp. 685–693, 2014.

Jain, P., Tewari, A., and Dhillon, I. S. Partial hard thresholding. *IEEE Transactions on Information Theory*, 63(5): 3029–3038, 2017.

Lewis, D. D., Yang, Y., Russell-Rose, T., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Liberty, E. and Sviridenko, M. Greedy minimization of weakly supermodular set functions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

Liu, H. and Foygel Barber, R. Between hard and soft thresholding: optimal iterative thresholding algorithms. *Information and Inference: A Journal of the IMA*, 9(4): 899–933, 2020.

Natarajan, B. K. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

Needell, D. and Tropp, J. A. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44. IEEE, 1993.

Peste, A., Iofinova, E., Vladu, A., and Alistarh, D. Ac/dc: Alternating compressed/decompressed training of deep neural networks. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Schmidt, L. *Algorithms above the noise floor*. PhD thesis, Massachusetts Institute of Technology, Cambridge, USA, 2018. URL http://hdl.handle.net/1721.1/118098.

Shalev-Shwartz, S., Srebro, N., and Zhang, T. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20(6): 2807–2832, 2010.

Shalev-Shwartz, S., Gonen, A., and Shamir, O. Large-scale convex minimization with a low-rank constraint. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 329–336, 2011.

Shen, J. and Li, P. On the iteration complexity of support recovery via hard thresholding pursuit. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3115–3124. JMLR. org, 2017a.

Shen, J. and Li, P. Partial hard thresholding: Towards a principled analysis of support recovery. In *Advances in Neural Information Processing Systems*, pp. 3124–3134, 2017b.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Yuan, X., Li, P., and Zhang, T. Exact recovery of hard thresholding pursuit. In *Advances in Neural Information Processing Systems*, pp. 3558–3566, 2016.

A. Python implementation

In Figure 2 we have python implementations of the IHT and regularized IHT algorithms that we use for our experiments. As can be seen, both implementations are pretty short.

B. Proof of Theorem 1.1

Theorem 1.1 (Regularized IHT)**.** Let $f \in \mathbb{R}^n \to \mathbb{R}$ be a convex function that is β-smooth and α-strongly convex, with condition number $\kappa = \beta/\alpha$, and $\boldsymbol{x}^*$ be an (unknown) s-sparse solution with support S^*. Then, running Algorithm 1 with $\eta = (2\beta)^{-1}$ and $c = s'/(4T)$ for

$$T = O\left(\kappa \log \frac{f(\boldsymbol{x}^0) + (\beta/2)\left\|\boldsymbol{x}^0\right\|_2^2 - f(\boldsymbol{x}^*)}{\varepsilon}\right)$$

iterations starting from an arbitrary $s' = O(s\kappa)$-sparse solution $\boldsymbol{x}^0$, the algorithm returns an s'-sparse solution $\boldsymbol{x}^T$ such that $f(\boldsymbol{x}^T) \leq f(\boldsymbol{x}^*) + \varepsilon$. Furthermore, each iteration requires $O(1)$ evaluations of f, ∇f, and $O(n)$ additional time.

Proof. We repeatedly apply Lemma 4.1 for

$$T = 64(\kappa+1)\log \frac{f(\boldsymbol{x}^0) + (\beta/2)\left\|\boldsymbol{x}^0\right\|_2^2 - f(\boldsymbol{x}^*)}{\varepsilon}$$

iterations. We define the regularized function

$$g^t(\boldsymbol{x}) := f(\boldsymbol{x}) + (\beta/2)\left\|\boldsymbol{x}\right\|_{\boldsymbol{w}^t,2}^2 ,$$

where $\boldsymbol{w}^t$ are the weights before iteration $t \in [0, T-1]$. Specifically, for each t we apply Lemma 4.1 on the current solution $\boldsymbol{x}^t$ and obtain the solution $\boldsymbol{x}^{t+1}$.

Before moving forward, we give an intuitive summary of the proof and the role of Lemma 4.1. As long as IHT makes "sufficient" progress on the regularized function g^t, this is satisfactory for the original function f as well, because $f(\boldsymbol{x}) \leq g^t(\boldsymbol{x})$ for all $\boldsymbol{x}$. This is the case of the first bullet of Lemma 4.1. If it stops making sufficient progress, this means we are at an (approximate) sparse optimum for g^t, although it is not necessarily a good sparse solution for f, which is the objective we are aiming to minimize. This is where the second bullet of Lemma 4.1 comes in, which gives necessary conditions for the above non-progress phenomenon (in other words, a partial characterization of the local minima encountered when running IHT on a regularized function). Specifically, the following condition is central to our approach:

$$\left\|\boldsymbol{x}^t_{S^*}\right\|_{(\boldsymbol{w}^t)^2,2}^2 \geq (4\kappa+6)^{-1}\left\|\boldsymbol{x}^t\right\|_{\boldsymbol{w}^t,2}^2 .$$

We use this condition in the second part of the proof (after Case 2) to motivate a weight update from $\boldsymbol{w}^t$ to $\boldsymbol{w}^{t+1}$, and

```python
import numpy as np

def IHT(n, s):
  x = np.zeros(n)
  for _ in range(T):
    x_new = x - eta * grad(x)
    x_new[np.argsort(np.abs(x_new))[:-s]] = 0
    x = x_new
  return x

def RegIHT(n, s):
  x, w = np.zeros(n), np.ones(n)
  for _ in range(T):
        x_new = (1 - 0.5 * w) * x - 0.5 * eta * grad(x)
        x_new[np.argsort(np.abs(x_new))[:-s]] = 0
        reg = np.sum(w * x**2)
        if reg != 0:
            w = w * (1 - c * w * x**2 / reg)
            w[w <= round_th] = 0
        x = x_new
  return x
```

Figure 2. Our python implementations of IHT, RegIHT, where grad is the gradient function, n is the total number of features, s is the desired sparsity level, eta is the step size, c is the weight step size, and round_th is the weight rounding threshold, which we set to 0.5. Note that grad(x) = np.dot(A.T, np.dot(A, x) - b) for linear regression and grad(x) = np.dot(A.T, expit(np.dot(A,x)) - b) for logistic regression, where expit is the sigmoid function.

show that, exactly because of this condition, a lot of the weight decrease is concentrated inside the optimal support S^*. As the total weight decrease in S^* is bounded by s, this gives a bound on the total number of iterations with insufficient decrease of g^t. If not for this condition, we would not be able to bound the number of such iterations and would have potentially remained forever stuck at a local minimum.

Now we are ready to move to the technical proof. In order to make sure that $g^{t+1}(\boldsymbol{x}^{t+1}) \leq g^{t+1}(\boldsymbol{x}^t)$, we revert to the previous solution if the one returned by Lemma 4.1 has a larger value of g^{t+1}. This is exactly what the conditional in Algorithm 1 is for. The property that $g^{t+1}(\boldsymbol{x}^{t+1}) \leq g^{t+1}(\boldsymbol{x}^t)$ is only used in the very last part of the proof.

Let us assume that $g^t(\boldsymbol{x}^t) > f(\boldsymbol{x}^*)$ at all times, as otherwise the statement holds by the fact that $g^t(\boldsymbol{x}^t)$ is non-increasing as a function of t and upper bounds $f(\boldsymbol{x}^t)$ for all t. We have $g^{t+1}(\boldsymbol{x}^{t+1}) \leq g^t(\boldsymbol{x}^t)$ by the fact that $\boldsymbol{w}^{t+1} \leq \boldsymbol{w}^t$ and $g^t(\boldsymbol{x}^{t+1}) \leq g^t(\boldsymbol{x}^t)$ by the guarantees of Lemma 4.1.

If the first bullet of Lemma 4.1 holds, we have that the value of g decreases considerably on iteration t, i.e.

$$\begin{aligned} &g^{t+1}(\boldsymbol{x}^{t+1}) \\ &\leq g^t(\boldsymbol{x}^{t+1}) \\ &\leq g^t(\boldsymbol{x}^t) - (16\kappa)^{-1}(g^t(\boldsymbol{x}^t) - f(\boldsymbol{x}^*))\,. \end{aligned}$$

Let us call these iterations *progress iterations*, and the other ones (where the second bullet of Lemma 4.1 holds) *weight iterations*. Now, since $g^t(\boldsymbol{x}^t)$ is non-increasing as a function of t, after $16\kappa \log \frac{g^0(\boldsymbol{x}^0) - f(\boldsymbol{x}^*)}{\varepsilon}$ progress iterations we will have

$$f(\boldsymbol{x}^t) \leq g^t(\boldsymbol{x}^t) \leq f(\boldsymbol{x}^*) + \varepsilon\,,$$

and so we will be done. From now on let us assume this is not the case, so there are at least

$$T - 16\kappa \log \frac{g^0(\boldsymbol{x}^0) - f(\boldsymbol{x}^*)}{\varepsilon} \geq 3T/4$$

weight iterations.

We remind that in each weight iteration, we have

$$\left\|\boldsymbol{x}^t_{S^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} \geq (4\kappa+6)^{-1}\left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^t,2} \tag{9}$$

$$(\beta/2)\left\|\boldsymbol{x}^t_{S^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} \geq (8\kappa+8)^{-1}\left(g^t(\boldsymbol{x}^t) - f(\boldsymbol{x}^*)\right)\,. \tag{10}$$

In words, (9) roughly implies that at least an $\Omega(1/\kappa)$ fraction of the mass of $(\boldsymbol{w}^t\boldsymbol{x}^t)^2$ lies inside S^*. Therefore, if we decrease $\boldsymbol{w}^t$ by a quantity proportional to $(\boldsymbol{w}^t\boldsymbol{x}^t)^2$, the total sum of weights will decrease at most $O(\kappa)$ times faster than the sum of weights inside S^*. As the latter quantity can only decrease by s overall, the total decrease of weights will be $O(s\kappa)$.

Concretely, after each iteration we update the regularization weights as follows:

$$\boldsymbol{w}^{t+1} = \left(\boldsymbol{w}^t - c\cdot(\boldsymbol{w}^t\boldsymbol{x}^t)^2/\left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^t,2}\right)_{\geq 1/2}\,,$$

for some $c > 0$ to be determined later. First of all, note that the weights are non-increasing. Now, if not for the thresholding operation, it is easy to see that the total weight decrease is at most c. The thresholding operation can only double this weight decrease to $2c$. Concretely, for all t we define a vector $\bar{\boldsymbol{w}}^t$ such that

$$\bar{w}_i^t = \begin{cases} w_i^t & \text{if } w_i^t \geq 1/2 \\ 1/2 & \text{if } w_i^t = 0\,. \end{cases}$$

Clearly, $\frac{1}{2}\left(\mathbf{1} - \boldsymbol{w}^t\right) \leq \mathbf{1} - \bar{\boldsymbol{w}}^t \leq \mathbf{1} - \boldsymbol{w}^t$. Now, we have

$$\begin{aligned} &\left\|\bar{\boldsymbol{w}}^t\right\|_1 - \left\|\bar{\boldsymbol{w}}^{t+1}\right\|_1 \\ &\leq c\left\|\boldsymbol{x}^t\right\|^2_{(\boldsymbol{w}^t)^2,2} / \left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^t,2} \\ &\leq c\,, \end{aligned}$$

and, summing up for all t we get

$$\left\|\mathbf{1} - \bar{\boldsymbol{w}}^T\right\|_1 = \left\|\bar{\boldsymbol{w}}^0\right\|_1 - \left\|\bar{\boldsymbol{w}}^T\right\|_1 \leq cT\,.$$

Therefore,

$$\left\|\mathbf{1} - \boldsymbol{w}^T\right\|_1 \leq 2\left\|\mathbf{1} - \bar{\boldsymbol{w}}^T\right\|_1 \leq 2cT\,,$$

and so $\left\|\boldsymbol{w}^T\right\|_1 \geq n - 2cT$.

Therefore, the condition $\left\|\boldsymbol{w}^t\right\|_1 \geq n - s'/2$ of Lemma 4.1 is satisfied for all t as long as $c \leq s'/(4T)$. In order to bound the number of iterations, we distinguish two cases for the sum of weights inside S^*.

Case 1: The sum of weights inside S^* decreases by $\geq 4s/T$.

This case cannot happen more than $T/4$ times since the sum of weights inside S^* can only decrease by s in total. Therefore, case 2 below happens at least $T/2$ times.

Case 2: The sum of weights inside S^* decreases by $< 4s/T$.

Note that the decrease in the sum of weights in S^* is exactly equal to

$$\sum_{i \in S^*} \begin{cases} c \cdot (w_i^t x_i^t)^2 / \left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^t,2} & \text{if this is} \leq w_i^t - 1/2 \\ w_i^t & \text{otherwise}\,. \end{cases}$$

Let T^* be the set of indices $i \in S^*$ for which the second case is true, i.e.

$$c \cdot (w_i^t x_i^t)^2 / \left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^t,2} > w_i^t - 1/2\,.$$

The total weight decrease from elements in $S^* \backslash T^*$ is then

$$\begin{aligned} &\sum_{i \in S^* \backslash T^*} c \cdot (w_i^t x_i^t)^2 / \left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^t,2} \\ &= c\left\|\boldsymbol{x}^t_{S^* \backslash T^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} / \left\|\boldsymbol{x}^t\right\|^2_{\boldsymbol{w}^t,2} \\ &\geq \frac{c}{4\kappa + 6}\left\|\boldsymbol{x}^t_{S^* \backslash T^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} / \left\|\boldsymbol{x}^t_{S^*}\right\|^2_{(\boldsymbol{w}^t)^2,2}\,, \end{aligned}$$

where we used (9). As we have assumed that this decrease is less than $4s/T$, we have that

$$\begin{aligned} \left\|\boldsymbol{x}^t_{T^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} &= \left\|\boldsymbol{x}^t_{S^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} - \left\|\boldsymbol{x}^t_{S^* \backslash T^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} \\ &\geq \left(1 - \frac{4s(4\kappa+6)}{cT}\right)\left\|\boldsymbol{x}^t_{S^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} \qquad (11) \\ &\geq (1/2)\left\|\boldsymbol{x}^t_{S^*}\right\|^2_{(\boldsymbol{w}^t)^2,2}\,, \end{aligned}$$

as long as $c \geq 8s(4\kappa+6)/T$. We can pick such a c as long as

$$8s(4\kappa+6)/T \leq c \leq s'/(4T) \Leftrightarrow s' \geq 32(4\kappa+6)s\,.$$

Now, to deal with the fact that the sum weights in T^* might not decrease sufficiently, note that all the weights in T^* are being set to 0, i.e. $w_i^{t+1} = 0$ for all $i \in T^*$. Together with (11) and (10) this means that we can make significant progress in function value. To see this, note that

$$\begin{aligned} &g^{t+1}(\boldsymbol{x}^{t+1}) \\ &\leq g^{t+1}(\boldsymbol{x}^t) \\ &\leq g^t(\boldsymbol{x}^t) - (\beta/2)\left\|\boldsymbol{x}^t_{T^*}\right\|^2_{\boldsymbol{w}^t,2} \\ &\leq g^t(\boldsymbol{x}^t) - (\beta/2)\left\|\boldsymbol{x}^t_{T^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} \\ &\leq g^t(\boldsymbol{x}^t) - (\beta/4)\left\|\boldsymbol{x}^t_{S^*}\right\|^2_{(\boldsymbol{w}^t)^2,2} \\ &\leq g^t(\boldsymbol{x}^t) - (16\kappa+16)^{-1}\left(g^t(\boldsymbol{x}^t) - f(\boldsymbol{x}^*)\right)\,, \end{aligned}$$

which can happen at most

$$16(\kappa+1)\log\frac{g^0(\boldsymbol{x}^0) - f(\boldsymbol{x}^*)}{\varepsilon} \leq T/4$$

times.

□

C. Proof of Lemma 4.1

Lemma 4.1 (Regularized IHT step progress)**.** *Let $f \in \mathbb{R}^n \to \mathbb{R}$ be a convex function that is β-smooth and α-strongly convex, $\kappa = \beta/\alpha$ be its condition number, and $\boldsymbol{x}^*$ be any s-sparse solution.*

Given any s'-sparse solution $\boldsymbol{x} \in \mathbb{R}^n$ where

$$s' \geq (128\kappa + 2)s$$

and a weight vector $\boldsymbol{w} \in (\{0\} \cup [1/2, 1])^n$ such that $\left\|\boldsymbol{w}\right\|_1 \geq n - s'/2$, we make the following update:

$$\boldsymbol{x}' = H_{s'}\left((\mathbf{1} - 0.5\boldsymbol{w})\boldsymbol{x} - (2\beta)^{-1}\nabla f(\boldsymbol{x})\right)\,.$$

Then, at least one of the following two conditions holds:

- *Updating $\boldsymbol{x}$ makes regularized progress:*

$$g(\boldsymbol{x}') \leq g(\boldsymbol{x}) - (16\kappa)^{-1}(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) ,$$

where

$$g(\boldsymbol{x}) := f(\boldsymbol{x}) + (\beta/2) \left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2$$

is the ℓ_2-regularized version of f with weights given by $\boldsymbol{w}$. Note: The regularized progress statement is true as long as $\boldsymbol{x}$ is suboptimal, i.e. $g(\boldsymbol{x}) > f(\boldsymbol{x}^*)$. Otherwise, we just have $g(\boldsymbol{x}') \leq g(\boldsymbol{x})$.

- *$\boldsymbol{x}$ is significantly correlated to the optimal support $S^* := \operatorname{supp}(\boldsymbol{x}^*)$:*

$$\left\|\boldsymbol{x}_{S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq (4\kappa + 6)^{-1} \left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2 ,$$

and the regularization term restricted to S^* is non-negligible:

$$(\beta/2) \left\|\boldsymbol{x}_{S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq (8\kappa + 8)^{-1} (g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) .$$

Proof. By using the fact that f is β-smooth, and so g is 2β-smooth due to $\boldsymbol{w} \leq \mathbf{1}$, for any $\boldsymbol{x}'$ we obtain

$$g(\boldsymbol{x}') - g(\boldsymbol{x}) \leq \langle \nabla g(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle + \beta \left\|\boldsymbol{x}' - \boldsymbol{x}\right\|_2^2 . \quad (12)$$

We let S be the support of $\boldsymbol{x}$ and S' the support of $\boldsymbol{x}'$, i.e. the set of s' indices of the largest magnitude entries of the vector. Since $\nabla g(\boldsymbol{x}) = \nabla f(\boldsymbol{x}) + \beta \boldsymbol{w}\boldsymbol{x}$, we have

$$\bar{\boldsymbol{x}} = \boldsymbol{x} - \eta \nabla g(\boldsymbol{x}) = (\mathbf{1} - 0.5\boldsymbol{w})\boldsymbol{x} - \eta \nabla f(\boldsymbol{x}) ,$$

where $\eta = (2\beta)^{-1}$. We let $A = S' \backslash S$ be the newly inserted entries and $B = S \backslash S'$ be the entries that were just removed from the support. Note that

$$\begin{aligned} \boldsymbol{x}' &= [\boldsymbol{x} - \eta \nabla g(\boldsymbol{x})]_{S'} \\ &= \boldsymbol{x} - \eta \nabla_{S'} g(\boldsymbol{x}) - \boldsymbol{x}_B \\ &= \boldsymbol{x} - \eta \nabla_{S' \cup B} g(\boldsymbol{x}) - \bar{\boldsymbol{x}}_B . \end{aligned}$$

Using (12), we have

$$\begin{aligned} & g(\boldsymbol{x}') - g(\boldsymbol{x}) \\ & \leq \langle \nabla g(\boldsymbol{x}), -\eta \nabla_{S' \cup B} g(\boldsymbol{x}) - \bar{\boldsymbol{x}}_B \rangle \\ & + \beta \left\| -\eta \nabla_{S' \cup B} g(\boldsymbol{x}) - \bar{\boldsymbol{x}}_B \right\|_2^2 \\ & = -(4\beta)^{-1} \left\| \nabla_{S' \cup B} g(\boldsymbol{x}) \right\|_2^2 + \beta \left\| \bar{\boldsymbol{x}}_B \right\|_2^2 \\ & = -\beta \left\| \eta \nabla_{S \cup A} g(\boldsymbol{x}) \right\|_2^2 + \beta \left\| \bar{\boldsymbol{x}}_B \right\|_2^2 \\ & \leq -\beta \left\| \eta \nabla_{S \cup A'} g(\boldsymbol{x}) \right\|_2^2 + \beta \left\| \bar{\boldsymbol{x}}_{B'} \right\|_2^2 , \end{aligned} \quad (13)$$

for any two sets $A' \in [n] \backslash S$ and $B' \subseteq S$ with $|A'| = |B'|$. The latter inequality follows because of the following lemma about IHT:

Lemma C.1. *Suppose that we run one step of IHT on vector $\boldsymbol{x}$ supported on S for some function g, and let the updated solution vector be $\boldsymbol{x}' = \bar{\boldsymbol{x}}_{(S \cup A) \backslash B}$, where $\bar{\boldsymbol{x}} = \boldsymbol{x} - \eta \nabla g(\boldsymbol{x})$. Then, for any $A' \subseteq [n] \backslash S$ and $B' \subseteq S$ with $|A'| = |B'|$, we have*

$$- \left\| \eta \nabla_A g(\boldsymbol{x}) \right\|_2^2 + \left\| \bar{\boldsymbol{x}}_B \right\|_2^2 \leq - \left\| \eta \nabla_{A'} g(\boldsymbol{x}) \right\|_2^2 + \left\| \bar{\boldsymbol{x}}_{B'} \right\|_2^2 . \quad (14)$$

Proof. If we denote $|A| = |B| = t$ and $|A'| = |B'| = t'$, then note that by definition of IHT, A are the t largest entries in

$$\left| \bar{\boldsymbol{x}}_{[n] \backslash S} \right| = \eta \left| \nabla_{[n] \backslash S} g(\boldsymbol{x}) \right| ,$$

and B are the t smallest entries in $|\bar{\boldsymbol{x}}_S|$. Similarly, we can assume that A' are the t' largest entries in $\eta \left| \nabla_{[n] \backslash S} g(\boldsymbol{x}) \right|$ and B' are the t' smallest entries in $|\bar{\boldsymbol{x}}_S|$, since this way the right hand side of (14) takes its minimum value. If $t' = t$, we are done. We consider two cases:

1. $t' > t$: In this case we have $A' \supseteq A$, $B' \supseteq B$, so

$$\begin{aligned} & - \left\| \eta \nabla_{A'} g(\boldsymbol{x}) \right\|_2^2 + \left\| \bar{\boldsymbol{x}}_{B'} \right\|_2^2 + \left\| \eta \nabla_A g(\boldsymbol{x}) \right\|_2^2 - \left\| \bar{\boldsymbol{x}}_B \right\|_2^2 \\ & = - \left\| \eta \nabla_{A' \backslash A} g(\boldsymbol{x}) \right\|_2^2 + \left\| \bar{\boldsymbol{x}}_{B' \backslash B} \right\|_2^2 \\ & = - \left\| \bar{\boldsymbol{x}}_{A' \backslash A} \right\|_2^2 + \left\| \bar{\boldsymbol{x}}_{B' \backslash B} \right\|_2^2 \\ & \geq (t' - t) \left(- \max_{i \in A' \backslash A} (\bar{x}_i)^2 + \min_{j \in B' \backslash B} (\bar{x}_j)^2 \right) \\ & \geq 0 , \end{aligned}$$

   where the last inequality follows since, by definition of the IHT step, $|\bar{x}_i| \leq |\bar{x}_j|$ for any $i \in A' \backslash A$ and $j \in B' \backslash B$. Otherwise, i would have taken j's place in S'.

2. $t' < t$: In this case we have $A' \subseteq A$, $B' \subseteq B$. Similarly to the previous case,

$$\begin{aligned} & - \left\| \eta \nabla_{A'} g(\boldsymbol{x}) \right\|_2^2 + \left\| \bar{\boldsymbol{x}}_{B'} \right\|_2^2 + \left\| \eta \nabla_A g(\boldsymbol{x}) \right\|_2^2 - \left\| \bar{\boldsymbol{x}}_B \right\|_2^2 \\ & = \left\| \eta \nabla_{A \backslash A'} g(\boldsymbol{x}) \right\|_2^2 - \left\| \bar{\boldsymbol{x}}_{B \backslash B'} \right\|_2^2 \\ & = \left\| \bar{\boldsymbol{x}}_{A \backslash A'} \right\|_2^2 - \left\| \bar{\boldsymbol{x}}_{B \backslash B'} \right\|_2^2 \\ & \geq (t - t') \left(\min_{i \in A \backslash A'} (\bar{x}_i)^2 - \max_{j \in B \backslash B'} (\bar{x}_j)^2 \right) \\ & \geq 0 , \end{aligned}$$

   where the last inequality follows since, by definition of the IHT step, $|\bar{x}_i| \geq |\bar{x}_j|$ for any $i \in A \backslash A'$ and $j \in B \backslash B'$. Otherwise i wouldn't have taken j's place in S'.

□

Now, let us assume that the first bullet in the lemma statement is false, i.e.

$$g(\boldsymbol{x}') - g(\boldsymbol{x}) > -(16\kappa)^{-1}(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) \,.$$

Setting $A' = B' = \emptyset$ in (13), we get that

$$g(\boldsymbol{x}') - g(\boldsymbol{x}) \leq -\beta \left\| \eta \nabla_S g(\boldsymbol{x}) \right\|_2^2 ,$$

so we conclude that

$$\|\nabla_S g(\boldsymbol{x})\|_2^2 < \frac{1}{16\kappa\beta\eta^2} \left(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)\right) = \frac{\alpha}{4} \left(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)\right) . \tag{15}$$

Now, we again use (13) but we set A' to be the s entries from $[n]\backslash S$ on which $\nabla g(\boldsymbol{x})$ has the largest magnitude, and B' to be the s entries from S on which $\bar{\boldsymbol{x}}$ has the smallest magnitude. Also, let R be an arbitrary subset of $S\backslash S^*$ with size $r \geq 2s$. We then have

$$\begin{aligned}
&g(\boldsymbol{x}') - g(\boldsymbol{x}) \\
&\leq -(4\beta)^{-1} \left\| \nabla_{S\cup A'} g(\boldsymbol{x}) \right\|_2^2 + \beta \left\| \bar{\boldsymbol{x}}_{B'} \right\|_2^2 \\
&\leq -(4\beta)^{-1} \left\| \nabla_{S\cup S^*} g(\boldsymbol{x}) \right\|_2^2 + \beta \left\| \bar{\boldsymbol{x}}_{B'} \right\|_2^2 \\
&\leq -(4\beta)^{-1} \left\| \nabla_{S\cup S^*} g(\boldsymbol{x}) \right\|_2^2 + \frac{\beta s}{|R\backslash S^*|} \left\| \bar{\boldsymbol{x}}_{R\backslash S^*} \right\|_2^2 \\
&\leq -(4\beta)^{-1} \left\| \nabla_{S\cup S^*} g(\boldsymbol{x}) \right\|_2^2 + \frac{\beta s}{r-s} \left\| \bar{\boldsymbol{x}}_{R\backslash S^*} \right\|_2^2 ,
\end{aligned} \tag{16}$$

where we used the fact that

$$\|\nabla_{A'} g(\boldsymbol{x})\|_2^2 \geq \left\| \nabla_{S^*\backslash S} g(\boldsymbol{x}) \right\|_2^2$$

by definition of A' (and since $|S^*\backslash S| \leq s$), and the fact that, by definition of B' (and since $|R\backslash S^*| \geq 2s - s = |B'|$),

$$\frac{1}{|B'|} \left\| \bar{\boldsymbol{x}}_{B'} \right\|_2^2 \leq \frac{1}{|R\backslash S^*|} \left\| \bar{\boldsymbol{x}}_{R\backslash S^*} \right\|_2^2 .$$

In fact, we will let $R = \{i \in S \mid w_i > 0\}$ be the set of elements that are being regularized. To lower bound the size r of this set, note that by the guarantee of the lemma statement,

$$\begin{aligned}
n - s'/2 &\leq \|\boldsymbol{w}\|_1 \\
&\leq n - |\{i \in S \mid w_i = 0\}| \\
&= n - (s' - r) ,
\end{aligned}$$

so $r \geq s'/2$. We conclude that $r \geq 2s$ since $s' \geq 4s$.

Now, because of the fact that f is α-strongly convex, we have

$$\begin{aligned}
&f(\boldsymbol{x}^*) - f(\boldsymbol{x}) \\
&\geq \langle \nabla f(\boldsymbol{x}), \boldsymbol{x}^* - \boldsymbol{x} \rangle + (\alpha/2) \left\| \boldsymbol{x}^* - \boldsymbol{x} \right\|_2^2 \\
&= \langle \nabla g(\boldsymbol{x}), \boldsymbol{x}^* - \boldsymbol{x} \rangle - \beta \langle \boldsymbol{w}\boldsymbol{x}, \boldsymbol{x}^* - \boldsymbol{x} \rangle + (\alpha/2) \left\| \boldsymbol{x}^* - \boldsymbol{x} \right\|_2^2 \\
&\geq -\alpha^{-1} \left\| \nabla_{S\cup S^*} g(\boldsymbol{x}) \right\|_2^2 \\
&\quad - \beta \langle \boldsymbol{w}\boldsymbol{x}, \boldsymbol{x}^* - \boldsymbol{x} \rangle + (\alpha/4) \left\| \boldsymbol{x}^* - \boldsymbol{x} \right\|_2^2 ,
\end{aligned} \tag{17}$$

where we used the inequality

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle + (\alpha/4) \left\| \boldsymbol{b} \right\|_2^2 \geq -\alpha^{-1} \left\| \boldsymbol{a} \right\|_2^2 .$$

By re-arranging and plugging (17) into (16), we get

$$\begin{aligned}
&g(\boldsymbol{x}') - g(\boldsymbol{x}) \\
&\leq -(4\kappa)^{-1} \Big(f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \\
&\quad - \beta \langle \boldsymbol{w}\boldsymbol{x}, \boldsymbol{x}^* - \boldsymbol{x} \rangle + (\alpha/4) \left\| \boldsymbol{x}^* - \boldsymbol{x} \right\|_2^2 \Big) \\
&\quad + \frac{\beta s}{r-s} \left\| \bar{\boldsymbol{x}}_{R\backslash S^*} \right\|_2^2 \\
&= -(4\kappa)^{-1} \Big(g(\boldsymbol{x}) - f(\boldsymbol{x}^*) \\
&\quad - \beta \langle \boldsymbol{w}\boldsymbol{x}, \boldsymbol{x}^* - \boldsymbol{x} \rangle - (\beta/2) \left\| \boldsymbol{x} \right\|_{\boldsymbol{w},2}^2 \\
&\quad + (\alpha/4) \left\| \boldsymbol{x}^* - \boldsymbol{x} \right\|_2^2 - \frac{4\kappa\beta s}{r-s} \left\| \bar{\boldsymbol{x}}_{R\backslash S^*} \right\|_2^2 \Big) .
\end{aligned} \tag{18}$$

Now, note that by definition of $\bar{\boldsymbol{x}}$ we have

$$\begin{aligned}
&\left\| \bar{\boldsymbol{x}}_{R\backslash S^*} \right\|_2^2 \\
&\leq 2 \left\| \boldsymbol{x}_{R\backslash S^*} \right\|_2^2 + 2(2\beta)^{-2} \left\| \nabla_{R\backslash S^*} g(\boldsymbol{x}) \right\|_2^2
\end{aligned}$$

and, since $w_i \geq 1/2$ for each $i \in R$,

$$\left\| \boldsymbol{x}_{R\backslash S^*} \right\|_2^2 \leq 2 \left\| \boldsymbol{x}_{R\backslash S^*} \right\|_{\boldsymbol{w},2}^2 \leq 2 \left\| \boldsymbol{x}_{S\backslash S^*} \right\|_{\boldsymbol{w},2}^2 .$$

Therefore,

$$\begin{aligned}
&\frac{4\kappa\beta s}{r-s} \left\| \bar{\boldsymbol{x}}_{R\backslash S^*} \right\|_2^2 \\
&\leq \frac{16\kappa\beta s}{r-s} \left\| \boldsymbol{x}_{S\backslash S^*} \right\|_{\boldsymbol{w},2}^2 + \frac{2\kappa s}{\beta(r-s)} \left\| \nabla_{R\backslash S^*} g(\boldsymbol{x}) \right\|_2^2 \\
&\leq \frac{16\kappa\beta s}{r-s} \left\| \boldsymbol{x}_{S\backslash S^*} \right\|_{\boldsymbol{w},2}^2 + \frac{s}{2(r-s)} (g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) ,
\end{aligned}$$

where the last inequality follows from (15) since $R\backslash S^* \subseteq S$.

Plugging this back into (18), we get

$$
\begin{aligned}
&g(\boldsymbol{x}') - g(\boldsymbol{x}) \\
&\leq -(4\kappa)^{-1}\Bigg(\left(1 - \frac{s}{2(r-s)}\right)(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) \\
&- \beta\langle \boldsymbol{w}\boldsymbol{x}, \boldsymbol{x}^* - \boldsymbol{x}\rangle - (\beta/2)\left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2 \\
&+ (\alpha/4)\left\|\boldsymbol{x}^* - \boldsymbol{x}\right\|_2^2 - \frac{16\kappa\beta s}{r-s}\left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2\Bigg) \\
&= -(4\kappa)^{-1}\Bigg(\left(1 - \frac{s}{2(r-s)}\right)(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) \\
&\underbrace{-\beta\langle \boldsymbol{w}\boldsymbol{x}_{S\cap S^*}, \boldsymbol{x}^* - \boldsymbol{x}\rangle + (\alpha/4)\left\|\boldsymbol{x}^* - \boldsymbol{x}\right\|_2^2}_{\geq -(\kappa\beta)\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2} \\
&+ \beta\left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2 - (\beta/2)\left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2 \underbrace{- \frac{16\kappa\beta s}{r-s}\left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2}_{\geq -(\beta/4)\left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2}\Bigg) \\
&\leq -(4\kappa)^{-1}\Bigg(\left(1 - \frac{s}{2(r-s)}\right)(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) \\
&- (\kappa\beta)\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 - (\beta/2)\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w},2}^2 \\
&+ (\beta/4)\left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2\Bigg) \\
&\leq -(4\kappa)^{-1}\Big(0.5\,(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) \\
&- (\kappa+1)\beta\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 + (\beta/4)\left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2\Big)\,,
\end{aligned}
\tag{19}
$$

where we used the fact that

$$
\frac{s}{2(r-s)} \leq 1/2\,,
$$

which holds as long as $s' \geq 4s$, and

$$
\frac{16\kappa\beta s}{r-s} \leq \beta/4\,,
$$

which holds as long as $r \geq s'/2$ and $s' \geq (128\kappa+2)s$. In the last inequality we also used the property $\boldsymbol{w}/2 \leq \boldsymbol{w}^2$, which is by definition of $\boldsymbol{w}$.

Now, note that, because we have assumed that the first bullet of the statement doesn't hold, it has to be the case that

$$
\begin{aligned}
&(1/4)\,(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) \\
&- (\kappa+1)\beta\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 + (\beta/4)\left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2 \leq 0\,.
\end{aligned}
$$

This immediately implies that

$$
\begin{aligned}
&(\kappa+1)\beta\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq (\beta/4)\left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2 \\
&\Rightarrow (4\kappa+4)\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq \left\|\boldsymbol{x}_{S\setminus S^*}\right\|_{\boldsymbol{w},2}^2 \\
&\Rightarrow (4\kappa+6)\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq \left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2\,,
\end{aligned}
$$

so

$$
\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq (4\kappa+6)^{-1}\left\|\boldsymbol{x}\right\|_{\boldsymbol{w},2}^2\,.
$$

Similarly we also have

$$
\begin{aligned}
&(\kappa+1)\beta\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq (1/4)\,(g(\boldsymbol{x}) - f(\boldsymbol{x}^*)) \\
&\Rightarrow (\beta/2)\left\|\boldsymbol{x}_{S\cap S^*}\right\|_{\boldsymbol{w}^2,2}^2 \geq (8\kappa+8)^{-1}(g(\boldsymbol{x}) - f(\boldsymbol{x}^*))\,.
\end{aligned}
$$

Therefore the second bullet of the statement is true, and we are done.

□

D. Low Rank Minimization

Algorithm 2 Regularized Local Search

$\boldsymbol{A}^0$: initial rank-r' solution
$\boldsymbol{W}^0 = \boldsymbol{Y}^0 = \boldsymbol{I}$: initial regularization weights
η: step size, T: #iterations
c: weight step size
for $t = 0, \ldots, T-1$ **do**
  $\Phi(\boldsymbol{A}) := (\beta/4)\left(\left\langle \boldsymbol{W}^t, \boldsymbol{A}\boldsymbol{A}^\top\right\rangle + \left\langle \boldsymbol{Y}^t, \boldsymbol{A}^\top\boldsymbol{A}\right\rangle\right)$
  $g(\boldsymbol{A}) := f(\boldsymbol{A}) + \Phi(\boldsymbol{A})$
  $\overline{\boldsymbol{A}} = H_{s'-1}\left(\boldsymbol{A}^t\right) - 0.5H_1\left(\eta\nabla g(\boldsymbol{A}^t)\right)$
  $\boldsymbol{P} = (\boldsymbol{W}^t)^{1/2}\boldsymbol{A}^t(\boldsymbol{A}^t)^\top(\boldsymbol{W}^t)^{1/2}$
  $\boldsymbol{Q} = (\boldsymbol{Y}^t)^{1/2}(\boldsymbol{A}^t)^\top\boldsymbol{A}^t(\boldsymbol{Y}^t)^{1/2}$
  $\Delta = g(\boldsymbol{A}^t) - f(\boldsymbol{A}^*)$

  if $g(\boldsymbol{A}^t) - g(\overline{\boldsymbol{A}}) \geq (r')^{-1}\Delta$ **then**
    Let $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be the SVD of $\overline{\boldsymbol{A}}$
    $\boldsymbol{A}^{t+1} = \underset{\boldsymbol{A}=\boldsymbol{U}\boldsymbol{X}\boldsymbol{V}^\top}{\operatorname{argmin}}\ g(\boldsymbol{A})$
    $\boldsymbol{W}^{t+1}, \boldsymbol{Y}^{t+1} = \boldsymbol{W}^t, \boldsymbol{Y}^t$

  else if

$$
\max\left\{\operatorname{Tr}\left[H_r\left(\boldsymbol{P}\right)\right], \operatorname{Tr}\left[H_r\left(\boldsymbol{Q}\right)\right]\right\} \geq (0.4/\beta)\Delta
$$

  then
    $\boldsymbol{A}^{t+1} = \boldsymbol{A}^t$
    $\boldsymbol{W}^{t+1} = (\boldsymbol{W}^t)^{1/2}\left(\boldsymbol{I} - r^{-1}\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{P})}\right)(\boldsymbol{W}^t)^{1/2}$
    $\boldsymbol{Y}^{t+1} = (\boldsymbol{Y}^t)^{1/2}\left(\boldsymbol{I} - r^{-1}\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{Q})}\right)(\boldsymbol{Y}^t)^{1/2}$

  else
    $\boldsymbol{A}^{t+1} = \boldsymbol{A}^t$
    $\boldsymbol{W}^{t+1} = \boldsymbol{W}^t - \frac{\boldsymbol{W}^t\boldsymbol{A}^t(\boldsymbol{A}^t)^\top\boldsymbol{W}^t}{\langle \boldsymbol{W}^t, \boldsymbol{A}^t(\boldsymbol{A}^t)^\top\rangle}$
    $\boldsymbol{Y}^{t+1} = \boldsymbol{Y}^t - \frac{\boldsymbol{Y}^t(\boldsymbol{A}^t)^\top\boldsymbol{A}^t\boldsymbol{Y}^t}{\langle \boldsymbol{Y}^t, (\boldsymbol{A}^t)^\top\boldsymbol{A}^t\rangle}$
  end if
end for

D.1. Preliminaries

We will use the following simple lemma about Frobenius products between low-rank projections and symmetric PSD matrices. We remind the reader that $H_r\left(\boldsymbol{A}\right)$ is the matrix consisting of the top r components from the singular value decomposition of $\boldsymbol{A}$.

Lemma D.1. *For any two symmetric PSD matrices $\boldsymbol{\Pi}, \boldsymbol{A} \in \mathbb{R}^{n\times n}$, where $\operatorname{rank}(\boldsymbol{\Pi}) \le r$ and $\|\boldsymbol{\Pi}\|_2 \le 1$, we have that*

$$|\langle \boldsymbol{\Pi}, \boldsymbol{A}\rangle| \le \operatorname{Tr}\left[H_r(\boldsymbol{A})\right] .$$

Proof. We will use the following inequality for singular values

$$\sum_{i=1}^{k} \sigma_i(AB) \le \sum_{i=1}^{k} \sigma_i(A)\sigma_i(B)$$

for $k = 1, \dots, n$, $A, B \in \mathbb{R}^{n\times n}$ and $\sigma_1(A) \ge \cdots \ge \sigma_n(A)$ are singular values of matrix A (see page 177 in (Horn & Johnson, 1991)). Then

$$\begin{aligned}
|\langle \boldsymbol{\Pi}, \boldsymbol{A}\rangle| &= \operatorname{Tr}\left[\boldsymbol{\Pi}\boldsymbol{A}^\top\right] \\
&= \operatorname{Tr}\left[\boldsymbol{\Pi}\boldsymbol{A}\right] \\
&= \sum_{i=1}^{n} \sigma_i(\boldsymbol{\Pi}\boldsymbol{A}) \\
&\le \sum_{i=1}^{n} \sigma_i(\boldsymbol{\Pi})\sigma_i(\boldsymbol{A}) \\
&= \sum_{i=1}^{r} \sigma_i(\boldsymbol{\Pi})\sigma_i(\boldsymbol{A}) \\
&\le \sum_{i=1}^{r} \sigma_i(\boldsymbol{A}) \\
&= \operatorname{Tr}\left[H_r(\boldsymbol{A})\right] .
\end{aligned}$$

□

D.2. Analysis

This section is devoted to proving Theorem 1.2, which analyzes an algorithm for low rank optimization that uses adaptive regularization.

Theorem 1.2 (Adaptive Regularization for Low Rank Optimization)**.** Let $f \in \mathbb{R}^{m\times n} \to \mathbb{R}$ be a convex function with condition number κ and consider the low rank minimization problem

$$\min_{\operatorname{rank}(\boldsymbol{A})\le r} f(\boldsymbol{A}) . \tag{20}$$

For any error parameter $\varepsilon > 0$, there exists a polynomial time algorithm that returns a matrix $\boldsymbol{A}$ with $\operatorname{rank}(\boldsymbol{A}) \le O\left(r\left(\kappa + \log\frac{f(\boldsymbol{O})-f(\boldsymbol{A}^*)}{\varepsilon}\right)\right)$ and $f(\boldsymbol{A}) \le f(\boldsymbol{A}^*) + \varepsilon$, where $\boldsymbol{A}^*$ is any rank-r matrix.

Proof of Theorem 1.2. Let the smoothness and strong convexity parameters of f be β, α. We repeatedly apply Lemma D.2 $T \ge O\left(r\kappa \log\frac{f(\boldsymbol{A}^0)+(\beta/2)\left\|\boldsymbol{A}^0\right\|_F^2 - f(\boldsymbol{A}^*)}{\varepsilon}\right)$ times starting from solution $\boldsymbol{A}^0 = \boldsymbol{O}$ and weight matrices $\boldsymbol{W}^0 = \boldsymbol{I}$, $\boldsymbol{Y}^0 = \boldsymbol{I}$. Thus, we obtain solutions $\boldsymbol{A}^0, \dots, \boldsymbol{A}^T$, and weights $\boldsymbol{W}^0, \boldsymbol{W}^1, \dots, \boldsymbol{W}^T$ and $\boldsymbol{Y}^0, \boldsymbol{Y}^1, \dots, \boldsymbol{Y}^T$. We let

$$\begin{aligned}
&g^t(\boldsymbol{A}) \\
&= f(\boldsymbol{A}) + (\beta/4)\left(\langle \boldsymbol{W}^t, \boldsymbol{A}^t(\boldsymbol{A}^t)^\top\rangle + \langle \boldsymbol{Y}^t, (\boldsymbol{A}^t)^\top\boldsymbol{A}^\top\rangle\right)
\end{aligned}$$

be the regularized function at iteration t.

We denote by T_i the total number of iterations for which item $i \in \{1, 2, 3\}$ from the statement of Lemma D.2 holds.

Consider the T_2 iterations for which item 2 from the statement of Lemma D.2 holds. Without loss of generality, $\boldsymbol{W}$ is updated at least $T_2/2$ times. Letting $\boldsymbol{A}^* = \boldsymbol{U}^*\boldsymbol{\Sigma}^*\boldsymbol{V}^{*\top}$ be the singular value decomposition of $\boldsymbol{A}^*$, for each such iteration we have

$$\begin{aligned}
&\operatorname{Tr}\left[\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\boldsymbol{W}^{t+1}\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\right] \\
&\le \operatorname{Tr}\left[\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\boldsymbol{W}^{t}\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\right] - (10\kappa)^{-1} ,
\end{aligned}$$

and for all other types of iterations we have $\boldsymbol{W}^{t+1} \preceq \boldsymbol{W}^t$. Therefore,

$$\begin{aligned}
&\operatorname{Tr}\left[\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\boldsymbol{W}^{T}\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\right] \\
&\le \operatorname{Tr}\left[\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\boldsymbol{W}^{0}\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\right] - \frac{T_2}{2}(10\kappa)^{-1} .
\end{aligned}$$

However, note that by the guarantee of Lemma D.2 that $\boldsymbol{W}^T \succeq \boldsymbol{O}$, we have

$$\operatorname{Tr}\left[\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\boldsymbol{W}^{T}\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\right] \ge 0 ,$$

and because $\boldsymbol{W}^0 = \boldsymbol{I}$ we also know that

$$\operatorname{Tr}\left[\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\boldsymbol{W}^{0}\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\right] = \operatorname{Tr}\left[\boldsymbol{\Pi}_{\operatorname{im}(\boldsymbol{U}^*)}\right] \le r .$$

This implies that $T_2 \le 20\kappa r$.

Now, if $T_1 \ge 16r\kappa\log\frac{g^0(\boldsymbol{A}^0)-f(\boldsymbol{A}^*)}{\varepsilon}$, and since $g^t(\boldsymbol{A}^t)$ is non-increasing for all t, we have

$$\begin{aligned}
&g^T(\boldsymbol{A}^T) - f(\boldsymbol{A}^*) \\
&\le \left(1 - (16r\kappa)^{-1}\right)^{T_1}\left(g^0(\boldsymbol{A}^0) - f(\boldsymbol{A}^*)\right) \\
&\le \varepsilon ,
\end{aligned}$$

so $T_1 \le 16r\kappa\log\frac{g^0(\boldsymbol{A}^0)-f(\boldsymbol{A}^*)}{\varepsilon}$.

Similarly, if $T_3 \ge 10r\log\frac{g^0(\boldsymbol{A}^0)-f(\boldsymbol{A}^*)}{\varepsilon}$ we have

$$\begin{aligned}
&g^T(\boldsymbol{A}^T) - f(\boldsymbol{A}^*) \\
&\le \left(1 - (10r)^{-1}\right)^{T_4}\left(g^0(\boldsymbol{A}^0) - f(\boldsymbol{A}^*)\right) \\
&\le \varepsilon ,
\end{aligned}$$

so $T_3 \geq 10r \log \frac{g^0(\boldsymbol{A}^0)-f(\boldsymbol{A}^*)}{\varepsilon}$.

Overall, we have that the total number of iterations is

$$T = \sum T_i \leq 36r(\kappa+1)\log\frac{g^0(\boldsymbol{A}^0)-f(\boldsymbol{A}^*)}{\varepsilon}\,.$$

The only thing left is to ensure that the conditions

$$\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}^t\right] \leq r'/2$$
$$\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{Y}^t\right] \leq r'/2$$

of Lemma D.2 are satisfied for all t. By the guarantees of Lemma D.2, if one of items $2, 3$ holds, then

$$\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}^{t+1}\right] \leq \mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}^t\right]+1\,,$$

and if item 1 holds, then

$$\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}^{t+1}\right] = \mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}^t\right]\,.$$

As $\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}^0\right]=0$, we have

$$\begin{aligned}&\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}^T\right]\\ &\leq T_2+T_3\\ &\leq 20\kappa r + 10r\log\frac{g^0(\boldsymbol{A}^0)-f(\boldsymbol{A}^*)}{\varepsilon}\\ &\leq r'/2\,,\end{aligned}$$

where the last inequality holds as long as

$$r' \geq 20r\left(2\kappa+\log\frac{g^0(\boldsymbol{A}^0)-f(\boldsymbol{A}^*)}{\varepsilon}\right)\,.$$

□

Lemma D.2 (Low rank minimization step analysis)**.** *Let $f:\mathbb{R}^{m\times n}\to\mathbb{R}$ be a β-smooth and α-strongly convex function with condition number $\kappa=\beta/\alpha$, and $\boldsymbol{W}\in\mathbb{R}^{m\times m}$, $\boldsymbol{Y}\in\mathbb{R}^{n\times n}$ be symmetric positive semi-definite weight matrices with spectral norm bounded by 1 and such that $\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}\right]\leq r'/2$ and $\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{Y}\right]\leq r'/2$ for fixed parameter $r'\geq 256r$. We define the regularized function*

$$g(\boldsymbol{A}) := f(\boldsymbol{A}) + \underbrace{(\beta/4)\left(\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle + \langle \boldsymbol{Y}, \boldsymbol{A}^\top\boldsymbol{A}\rangle\right)}_{\Phi(\boldsymbol{A})}\,.$$

Now, consider a rank-r' matrix $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ with singular value decomposition

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top = \sum_{j\in S}\lambda_j \boldsymbol{u}_j\boldsymbol{v}_j^\top$$

and with the property that

$$\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\cdot\nabla g(\boldsymbol{A})\cdot\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})} = \boldsymbol{O}\,.$$

For any rank-r solution $\boldsymbol{A}^*$ where $r'\geq 256r$, there is a procedure that updates $\boldsymbol{A}$, $\boldsymbol{W}$, $\boldsymbol{Y}$, and for which exactly one of the following scenarios holds:

1. *$\boldsymbol{A}$ is updated to a rank-r' matrix $\boldsymbol{A}'$, and $\boldsymbol{W}$, $\boldsymbol{Y}$ are not updated. We have sufficient progress in the regularized function:*

$$g(\boldsymbol{A}') \leq g(\boldsymbol{A}) - (16\kappa r)^{-1}\left(g(\boldsymbol{A})-f(\boldsymbol{A}^*)\right)\,.$$

2. *Exactly one of $\boldsymbol{W}$ or $\boldsymbol{Y}$ is updated (wlog $\boldsymbol{W}$) to a symmetric PSD $\boldsymbol{W}'\preceq\boldsymbol{W}$, and $\boldsymbol{A}$ is not updated. We have*

$$\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}'\right] \leq \mathrm{Tr}[\boldsymbol{I}-\boldsymbol{W}]+1$$

and

$$\begin{aligned}&\mathrm{Tr}\left[\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}\boldsymbol{W}'\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}\right]\\ &\leq \mathrm{Tr}\left[\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}\boldsymbol{W}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}\right] - (10\kappa)^{-1}\,.\end{aligned}$$

Respectively, for $\boldsymbol{Y}$:

$$\begin{aligned}&\mathrm{Tr}\left[\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}\boldsymbol{Y}'\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}\right]\\ &\leq \mathrm{Tr}\left[\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}\boldsymbol{Y}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}\right] - (10\kappa)^{-1}\,.\end{aligned}$$

3. *Exactly one of $\boldsymbol{W}$ or $\boldsymbol{Y}$ is updated (wlog $\boldsymbol{W}$) to a symmetric PSD $\boldsymbol{W}'\preceq\boldsymbol{W}$, and $\boldsymbol{A}$ is not updated. We have sufficient progress in the regularized function, where g' is the regularized function with the new weights:*

$$g'(\boldsymbol{A}) \leq g(\boldsymbol{A}) - (10r)^{-1}\left(g(\boldsymbol{A})-f(\boldsymbol{A}^*)\right)\,.$$

Additionally,

$$\mathrm{Tr}\left[\boldsymbol{I}-\boldsymbol{W}'\right] \leq \mathrm{Tr}[\boldsymbol{I}-\boldsymbol{W}]+1$$

Proof. We attempt to make the update $\boldsymbol{A}\to\boldsymbol{A}'$ as defined in Lemma D.3. If it makes enough progress, i.e.

$$g(\boldsymbol{A}') \leq g(\boldsymbol{A}) - (16\kappa r)^{-1}\left(g(\boldsymbol{A})-f(\boldsymbol{A}^*)\right)\,,$$

we are done. Otherwise, one of the items 2-5 in the statement of Lemma D.3 must hold. Let us take them one by one.

Item 2:

$$\langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}, \boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}\rangle \geq (10\kappa)^{-1}\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle\,.$$

We update $\boldsymbol{W}$ as

$$\boldsymbol{W}' = \boldsymbol{W} - c\cdot\boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}\,,$$

where $c=\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle^{-1}$. Note that this update preserves symmetry, and

$$\boldsymbol{O}\preceq\boldsymbol{W}'\preceq\boldsymbol{W}\,.$$

This is because

$$c\boldsymbol{W}^{1/2}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}^{1/2} \preceq c\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle \cdot \boldsymbol{I} \preceq \boldsymbol{I}\,,$$

so

$$\boldsymbol{W}' = \boldsymbol{W}^{1/2}\left(\boldsymbol{I} - c\boldsymbol{W}^{1/2}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}^{1/2}\right)\boldsymbol{W}^{1/2} \succeq \boldsymbol{O}$$

and

$$\boldsymbol{W}' = \boldsymbol{W} - c\boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W} \preceq \boldsymbol{W}\,.$$

Now, note that

$$\begin{aligned}\operatorname{Tr}\left[\boldsymbol{I} - \boldsymbol{W}'\right] &= \operatorname{Tr}\left[\boldsymbol{I} - \boldsymbol{W}\right] + c\langle \boldsymbol{W}^2, \boldsymbol{A}\boldsymbol{A}^\top\rangle \\ &\le \operatorname{Tr}\left[\boldsymbol{I} - \boldsymbol{W}\right] + c\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle \\ &= \operatorname{Tr}\left[\boldsymbol{I} - \boldsymbol{W}\right] + 1\,,\end{aligned}$$

where we used the fact that $\boldsymbol{W}^2 \preceq \boldsymbol{W}$, and (letting $\boldsymbol{\Pi}^* = \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}$ for convenience),

$$\begin{aligned}\operatorname{Tr}\left[\boldsymbol{\Pi}^*\boldsymbol{W}'\boldsymbol{\Pi}^*\right] &= \operatorname{Tr}\left[\boldsymbol{\Pi}^*\boldsymbol{W}\boldsymbol{\Pi}^*\right] - c\langle \boldsymbol{\Pi}^*, \boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}\rangle \\ &\le \operatorname{Tr}\left[\boldsymbol{\Pi}^*\boldsymbol{W}\boldsymbol{\Pi}^*\right] - c/(10\kappa)\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle \\ &= \operatorname{Tr}\left[\boldsymbol{\Pi}^*\boldsymbol{W}\boldsymbol{\Pi}^*\right] - (10\kappa)^{-1}\,,\end{aligned} \tag{21}$$

Item 3:

$$\langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}, \boldsymbol{Y}\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Y}\rangle \ge (10\kappa)^{-1}\langle \boldsymbol{Y}, \boldsymbol{A}^\top\boldsymbol{A}\rangle\,.$$

This is entirely analogous to the previous case.

Item 4:

$$(\beta/4)\operatorname{Tr}\left[H_r\left(\boldsymbol{A}^\top\boldsymbol{W}\boldsymbol{A}\right)\right] \ge 10^{-1}\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right)\,. \tag{22}$$

After considering the eigendecomposition

$$\boldsymbol{W}^{1/2}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}^{1/2} = \sum_{i\in[r']}\bar{\lambda}_i\bar{\boldsymbol{v}}_i\bar{\boldsymbol{v}}_i^\top$$

with $\bar{\lambda}_1 \ge \bar{\lambda}_2 \ge \cdots \ge \bar{\lambda}_{r'} \ge 0$, (22) can be re-phrased as

$$(\beta/4)\sum_{i\in[r]}\bar{\lambda}_i > (1/10)\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right)\,.$$

We update $\boldsymbol{W}$ as

$$\boldsymbol{W}' = \boldsymbol{W}^{1/2}\left(\boldsymbol{I} - r^{-1}\sum_{i\in[r]}\bar{\boldsymbol{v}}_i\bar{\boldsymbol{v}}_i^\top\right)\boldsymbol{W}^{1/2}$$

and let g' be the new regularized objective. First of all, note that this operation preserves symmetry, and that $\boldsymbol{O} \preceq \boldsymbol{W}' \preceq \boldsymbol{I}$, since $\sum_{i\in[r]}\bar{\boldsymbol{v}}_i\bar{\boldsymbol{v}}_i^\top \preceq \boldsymbol{I}$. Additionally,

$$\begin{aligned}\operatorname{Tr}\left[\boldsymbol{I} - \boldsymbol{W}'\right] &= \operatorname{Tr}\left[\boldsymbol{I} - \boldsymbol{W}\right] + r^{-1}\sum_{i\in[r]}\bar{\boldsymbol{v}}_i^\top\boldsymbol{W}\bar{\boldsymbol{v}}_i \\ &\le \operatorname{Tr}\left[\boldsymbol{W}\right] + 1\end{aligned}$$

and

$$\begin{aligned}&g'(\boldsymbol{A}) - g(\boldsymbol{A}) \\ &= (\beta/4)\langle \boldsymbol{W}', \boldsymbol{A}\boldsymbol{A}^\top\rangle - (\beta/4)\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle \\ &= -(\beta/(4r))\left\langle \boldsymbol{W}^{1/2}\left(\sum_{i\in[r]}\bar{\boldsymbol{v}}_i\bar{\boldsymbol{v}}_i^\top\right)\boldsymbol{W}^{1/2}, \boldsymbol{A}\boldsymbol{A}^\top\right\rangle \\ &= -(\beta/(4r))\sum_{i\in[r]}\bar{\lambda}_i \\ &\le -(10r)^{-1}(g(\boldsymbol{A}) - f(\boldsymbol{A}^*))\,,\end{aligned}$$

Item 5:

$$(\beta/4)\operatorname{Tr}\left[H_r\left(\boldsymbol{A}\boldsymbol{Y}\boldsymbol{A}^\top\right)\right] \ge 10^{-1}\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right)\,.$$

This is entirely analogous to the previous case. □

Lemma D.3. *Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a β-smooth and α-strongly convex function with condition number $\kappa = \beta/\alpha$, and $\boldsymbol{W} \in \mathbb{R}^{m\times m}$, $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ be symmetric positive semi-definite weight matrices with spectral norm bounded by 1 and such that $\operatorname{Tr}\left[\boldsymbol{I} - \boldsymbol{W}\right], \operatorname{Tr}\left[\boldsymbol{I} - \boldsymbol{Y}\right] \le r'/2$ for some parameter $r' \ge 0$. We define the regularized function*

$$g(\boldsymbol{A}) := f(\boldsymbol{A}) + \underbrace{(\beta/4)\left(\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle + \langle \boldsymbol{Y}, \boldsymbol{A}^\top\boldsymbol{A}\rangle\right)}_{\Phi(\boldsymbol{A})}\,.$$

Now, consider a rank-r' matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with singular value decomposition

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top = \sum_{j\in S}\lambda_j\boldsymbol{u}_j\boldsymbol{v}_j^\top$$

and with the property that

$$\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\cdot\nabla g(\boldsymbol{A})\cdot\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})} = \boldsymbol{O}\,.$$

We define an updated solution

$$\boldsymbol{A}' = \boldsymbol{A} - \eta\cdot H_1(\nabla g(\boldsymbol{A})) - \lambda_j\boldsymbol{u}_j\boldsymbol{v}_j^\top\,,$$

where $\eta = (2\beta)^{-1}$, $H_1(\cdot)$ returns the top singular component, and $j \in S$ is picked to minimize λ_j.

Then, for any rank-r solution $\boldsymbol{A}^*$, where $r' \ge 256r$, and its singular value decomposition $\boldsymbol{A}^* = \boldsymbol{U}^*\boldsymbol{\Lambda}^*\boldsymbol{V}^{*\top}$, at least one of the following conditions holds:

1. *We have sufficient progress in the regularized function:*

$$g(\boldsymbol{A}') \le g(\boldsymbol{A}) - (16\kappa r)^{-1}\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right)\,.$$

2. *$\boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}$ is significantly correlated to $\boldsymbol{U}^*$:*

$$\langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}, \boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{W}\rangle \ge (10\kappa)^{-1}\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle\,.$$

3. *$\boldsymbol{Y}\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Y}$ is significantly correlated to $\boldsymbol{V}^*$:*

$$\langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}, \boldsymbol{Y}\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Y}\rangle \geq (10\kappa)^{-1}\langle \boldsymbol{Y}, \boldsymbol{A}^\top\boldsymbol{A}\rangle .$$

4. *The spectrum of $\boldsymbol{A}^\top\boldsymbol{W}\boldsymbol{A}$ is highly concentrated and responsible for a constant fraction of the error:*

$$(\beta/4)\,\mathrm{Tr}\left[H_r\left(\boldsymbol{A}^\top\boldsymbol{W}\boldsymbol{A}\right)\right] \geq 10^{-1}\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right) .$$

and

5. *The spectrum of $\boldsymbol{A}\boldsymbol{Y}\boldsymbol{A}^\top$ is highly concentrated and responsible for a constant fraction of the error:*

$$(\beta/4)\,\mathrm{Tr}\left[H_r\left(\boldsymbol{A}\boldsymbol{Y}\boldsymbol{A}^\top\right)\right] \geq 10^{-1}\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right) .$$

Proof. Note that g is a 2β-smooth function. This follows because

$$\nabla g(\boldsymbol{A}) = \nabla f(\boldsymbol{A}) + (\beta/2)\left(\boldsymbol{W}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{Y}\right) ,$$

and so for any two matrices $\boldsymbol{A}, \boldsymbol{A}'$,

$$\begin{aligned}
&\left\|\nabla g(\boldsymbol{A}') - \nabla g(\boldsymbol{A})\right\|_F \\
&\leq \left\|\nabla f(\boldsymbol{A}') - \nabla f(\boldsymbol{A})\right\|_F \\
&+ (\beta/2)\left\|\boldsymbol{W}(\boldsymbol{A}' - \boldsymbol{A})\right\|_F + (\beta/2)\left\|(\boldsymbol{A}' - \boldsymbol{A})\boldsymbol{Y}\right\|_F \\
&\leq 2\beta\left\|\boldsymbol{A}' - \boldsymbol{A}\right\|_F ,
\end{aligned}$$

which is known to imply 2β-smoothness of g. Here we used the triangle inequality and the fact that $\boldsymbol{W}, \boldsymbol{Y} \preceq \boldsymbol{I}$. Therefore, we have

$$\begin{aligned}
&g(\boldsymbol{A}') - g(\boldsymbol{A}) \\
&\leq \langle \nabla g(\boldsymbol{A}), \boldsymbol{A}' - \boldsymbol{A}\rangle + \left\|\nabla g(\boldsymbol{A}') - \nabla g(\boldsymbol{A})\right\|_F \left\|\boldsymbol{A}' - \boldsymbol{A}\right\|_F \\
&\leq \langle \nabla g(\boldsymbol{A}), \boldsymbol{A}' - \boldsymbol{A}\rangle + \beta\left\|\boldsymbol{A}' - \boldsymbol{A}\right\|_F^2 \\
&\leq -\eta\left\|\nabla g(\boldsymbol{A})\right\|_2^2 + 2\beta\eta^2\left\|\nabla g(\boldsymbol{A})\right\|_2^2 + 2\beta\lambda_j^2 \\
&= -(8\beta)^{-1}\left\|\nabla g(\boldsymbol{A})\right\|_2^2 + 2\beta\lambda_j^2 ,
\end{aligned} \tag{23}$$

where in the second inequality we used the facts that

$$\begin{aligned}
&\langle \nabla g(\boldsymbol{A}), -\lambda_j \boldsymbol{u}_j\boldsymbol{v}_j^\top\rangle \\
&= \langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\nabla g(\boldsymbol{A})\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}, -\lambda_j \boldsymbol{u}_j\boldsymbol{v}_j^\top\rangle \\
&= 0
\end{aligned}$$

and that, for any two matrices $\boldsymbol{B}, \boldsymbol{C}$,

$$\left\|\boldsymbol{B} + \boldsymbol{C}\right\|_F^2 \leq 2\left\|\boldsymbol{B}\right\| + 2\left\|\boldsymbol{C}\right\|_F^2 .$$

The last equality follows by our choice of η. In order to lower bound $\left\|\nabla g(\boldsymbol{A})\right\|_2^2$, we use the strong convexity of f as follows:

$$\begin{aligned}
&f(\boldsymbol{A}^*) - f(\boldsymbol{A}) \\
&\geq \langle \nabla f(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A}\rangle + (\alpha/2)\left\|\boldsymbol{A}^* - \boldsymbol{A}\right\|_F^2 \\
&= \langle \nabla g(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A}\rangle \\
&- \langle \nabla \Phi(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A}\rangle + (\alpha/2)\left\|\boldsymbol{A}^* - \boldsymbol{A}\right\|_F^2 \\
&= \underbrace{\langle \nabla g(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A}\rangle + (\alpha/4)\left\|\boldsymbol{A}^* - \boldsymbol{A}\right\|_F^2}_{P} \\
&- \langle \nabla \Phi(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A}\rangle + (\alpha/4)\left\|\boldsymbol{A}^* - \boldsymbol{A}\right\|_F^2 .
\end{aligned} \tag{24}$$

Bounding P. We let $\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}$ be the orthogonal projections onto the images of $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively, so we can write

$$\begin{aligned}
&\boldsymbol{A}^* - \boldsymbol{A} \\
&= \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\left(\boldsymbol{A}^* - \boldsymbol{A}\right)\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})} \\
&+ \left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\right)\left(\boldsymbol{A}^* - \boldsymbol{A}\right)\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})} \\
&+ \left(\boldsymbol{A}^* - \boldsymbol{A}\right)\left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\right) \\
&= \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\left(\boldsymbol{A}^* - \boldsymbol{A}\right)\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})} \\
&+ \left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\right)\boldsymbol{A}^*\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})} \\
&+ \boldsymbol{A}^*\left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\right) .
\end{aligned}$$

Now, note that

$$\begin{aligned}
&\langle \nabla g(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A}\rangle \\
&= \langle \nabla g(\boldsymbol{A}), \left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\right)\boldsymbol{A}^*\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\rangle \\
&+ \langle \nabla g(\boldsymbol{A}), \boldsymbol{A}^*\left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\right)\rangle ,
\end{aligned}$$

where we used the fact that

$$\begin{aligned}
&\langle \nabla g(\boldsymbol{A}), \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}(\boldsymbol{A}^* - \boldsymbol{A})\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\rangle \\
&= \langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\nabla g(\boldsymbol{A})\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}, \boldsymbol{A}^* - \boldsymbol{A}\rangle \\
&= 0 ,
\end{aligned}$$

and

$$\begin{aligned}
&(\alpha/4)\left\|\boldsymbol{A}^* - \boldsymbol{A}\right\|_F^2 \\
&\geq (\alpha/4)\left\|\left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\right)\boldsymbol{A}^*\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\right\|_F^2 \\
&+ (\alpha/4)\left\|\boldsymbol{A}^*\left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\right)\right\|_F^2 .
\end{aligned}$$

Additionally, note that for any rank-r matrix $\boldsymbol{B}$, we have

$$\begin{aligned}
&\langle \nabla g(\boldsymbol{A}), \boldsymbol{B}\rangle + (\alpha/4)\left\|\boldsymbol{B}\right\|_F^2 \\
&\geq -\alpha^{-1}\left\|H_r\left(\nabla g(\boldsymbol{A})\right)\right\|_F^2 \\
&\geq -\alpha^{-1}r\left\|\nabla g(\boldsymbol{A})\right\|_2^2 ,
\end{aligned}$$

a proof of which can be found e.g. in Lemma A.6 of (Axiotis & Sviridenko, 2021a). Applying this inequality with

$$\boldsymbol{B} = \left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U})}\right)\boldsymbol{A}^*\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}$$

and

$$\boldsymbol{B} = \boldsymbol{A}^* \left(\boldsymbol{I} - \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V})}\right)$$

and summing them up, we obtain

$$\begin{aligned} P &= \langle \nabla g(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A} \rangle + (\alpha/4) \left\| \boldsymbol{A}^* - \boldsymbol{A} \right\|_F^2 \\ &\geq -2\alpha^{-1} r \left\| \nabla g(\boldsymbol{A}) \right\|_2^2 . \end{aligned}$$

Plugging this into (24) and re-arranging, we get

$$\begin{aligned} &\left\| \nabla g(\boldsymbol{A}) \right\|_2^2 \\ &\geq \alpha/(2r) \Big(f(\boldsymbol{A}) - f(\boldsymbol{A}^*) \\ &- \langle \nabla \Phi(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A} \rangle + (\alpha/4) \left\| \boldsymbol{A}^* - \boldsymbol{A} \right\|_F^2 \Big) \\ &= \alpha/(2r) \Big(g(\boldsymbol{A}) - f(\boldsymbol{A}^*) - \Phi(\boldsymbol{A}) \\ &\underbrace{- \langle \nabla \Phi(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A} \rangle + (\alpha/4) \left\| \boldsymbol{A}^* - \boldsymbol{A} \right\|_F^2}_{Q} \Big) . \end{aligned} \tag{25}$$

Bounding Q. We know that

$$-\langle \nabla \Phi(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A} \rangle = -(\beta/2) \langle \boldsymbol{W}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{Y}, \boldsymbol{A}^* - \boldsymbol{A} \rangle .$$

If we let

$$\boldsymbol{A}^* = \boldsymbol{U}^* \boldsymbol{\Lambda}^* \boldsymbol{V}^{*\top}$$

be the SVD of $\boldsymbol{A}^*$ and $\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}$, $\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}$ be the orthogonal projections onto the images of $\boldsymbol{U}^*$ and $\boldsymbol{V}^*$ respectively, then we have

$$\begin{aligned} &- (\beta/2) \langle \boldsymbol{W}\boldsymbol{A}, \boldsymbol{A}^* - \boldsymbol{A} \rangle \\ &= -(\beta/2) \langle \boldsymbol{W}\boldsymbol{A}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} (\boldsymbol{A}^* - \boldsymbol{A}) \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle \\ &+ (\beta/2) \langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top \rangle - (\beta/2) \langle \boldsymbol{W}\boldsymbol{A}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle . \end{aligned}$$

Looking at the first term of this, we have

$$\begin{aligned} &- (\beta/2) \langle \boldsymbol{W}\boldsymbol{A}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} (\boldsymbol{A}^* - \boldsymbol{A}) \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle \\ &+ (\alpha/8) \left\| \boldsymbol{A}^* - \boldsymbol{A} \right\|_F^2 \\ &= -(\beta/2) \langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{W}\boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}, \boldsymbol{A}^* - \boldsymbol{A} \rangle \\ &+ (\alpha/8) \left\| \boldsymbol{A}^* - \boldsymbol{A} \right\|_F^2 \\ &\geq -\beta^2/(2\alpha) \left\| \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{W}\boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \right\|_F^2 . \end{aligned}$$

Similarly for the terms containing $\boldsymbol{Y}$, we get

$$\begin{aligned} &- (\beta/2) \langle \boldsymbol{A}\boldsymbol{Y}, \boldsymbol{A}^* - \boldsymbol{A} \rangle \\ &= -(\beta/2) \langle \boldsymbol{A}\boldsymbol{Y}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} (\boldsymbol{A}^* - \boldsymbol{A}) \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle \\ &+ (\beta/2) \langle \boldsymbol{Y}, \boldsymbol{A}^\top \boldsymbol{A} \rangle - (\beta/2) \langle \boldsymbol{A}\boldsymbol{Y}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle . \end{aligned}$$

and

$$\begin{aligned} &- (\beta/2) \langle \boldsymbol{A}\boldsymbol{Y}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} (\boldsymbol{A}^* - \boldsymbol{A}) \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle \\ &+ (\alpha/8) \left\| \boldsymbol{A}^* - \boldsymbol{A} \right\|_F^2 \\ &\geq -\beta^2/(2\alpha) \left\| \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{A}\boldsymbol{Y} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \right\|_F^2 . \end{aligned}$$

In summary, we have

$$\begin{aligned} Q &= -\langle \nabla \Phi(\boldsymbol{A}), \boldsymbol{A}^* - \boldsymbol{A} \rangle + (\alpha/4) \left\| \boldsymbol{A}^* - \boldsymbol{A} \right\|_F^2 \\ &\geq -\beta^2/(2\alpha) \left\| \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{W}\boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \right\|_F^2 \\ &- \beta^2/(2\alpha) \left\| \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{A}\boldsymbol{Y} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \right\|_F^2 \\ &+ (\beta/2) \langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top \rangle + (\beta/2) \langle \boldsymbol{Y}, \boldsymbol{A}^\top \boldsymbol{A} \rangle \\ &- (\beta/2) \langle \boldsymbol{W}\boldsymbol{A}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle \\ &- (\beta/2) \langle \boldsymbol{A}\boldsymbol{Y}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle . \end{aligned} \tag{26}$$

Now, let us assume that all items 2-5 from the lemma statement are false. For the first term of (26), we have

$$\begin{aligned} &- \beta^2/(2\alpha) \left\| \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{W}\boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \right\|_F^2 \\ &\geq -\beta^2/(2\alpha) \left\| \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{W}\boldsymbol{A} \right\|_F^2 \\ &= -\beta^2/(2\alpha) \langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}, \boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{W} \rangle \\ &\geq -(\beta/20) \langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top \rangle , \end{aligned}$$

where we used item 2 from the lemma statement, and similarly for the second term of (26),

$$\begin{aligned} &- \beta^2/(2\alpha) \left\| \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{A}\boldsymbol{Y} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \right\|_F^2 \\ &\geq -(\beta/20) \langle \boldsymbol{Y}, \boldsymbol{A}^\top \boldsymbol{A} \rangle . \end{aligned}$$

Now we look at the second to last term of (26), i.e.

$$\begin{aligned} &- (\beta/2) \langle \boldsymbol{W}\boldsymbol{A}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \boldsymbol{A} \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \rangle \\ &= -(\beta/2) \langle \boldsymbol{W}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \boldsymbol{A}^\top, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \rangle . \end{aligned}$$

Now, we use the matrix Holder inequality

$$\begin{aligned} &- (\beta/2) \langle \boldsymbol{W}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \boldsymbol{A}^\top, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \rangle \\ &\geq -(\beta/2) \left\| \boldsymbol{W}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \boldsymbol{A}^\top \right\|_* \left\| \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)} \right\|_2 \\ &\geq -(\beta/2) \left\| \boldsymbol{W}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \boldsymbol{A}^\top \right\|_* , \end{aligned}$$

which can be proved by applying von Neumann's trace inequality and then the classical Holder inequality. Now, note that the matrix $\boldsymbol{W}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}\boldsymbol{A}^\top$ is similar to $\boldsymbol{W}^{1/2}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}\boldsymbol{A}^\top \boldsymbol{W}^{1/2}$, and so they have the same eigenvalues. Furthermore, the latter is a symmetric PSD matrix, and so the former has real positive eigenvalues as well. This means that its singular values are the same as its eigenvalues, and as a result the nuclear norm is equal to the trace, i.e.

$$\begin{aligned} &- (\beta/2) \left\| \boldsymbol{W}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \boldsymbol{A}^\top \right\|_* \\ &= -(\beta/2) \operatorname{Tr} \left(\boldsymbol{W}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)} \boldsymbol{A}^\top \right) \\ &= -(\beta/2) \langle \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}, \boldsymbol{A}^\top \boldsymbol{W}\boldsymbol{A} \rangle \\ &\geq -(\beta/2) \operatorname{Tr} \left[H_r \left(\boldsymbol{A}^\top \boldsymbol{W}\boldsymbol{A} \right) \right] \\ &\geq -(1/5) \left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*) \right) . \end{aligned}$$

where we also used Lemma D.1 and item 4 from the lemma statement. So we derived that

$$\begin{aligned}&-(\beta/2)\langle \boldsymbol{W}\boldsymbol{A}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}\rangle\\ &\geq -(1/5)\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right),\end{aligned}$$

and similarly for the last term of (26),

$$\begin{aligned}&-(\beta/2)\langle \boldsymbol{A}\boldsymbol{Y}, \boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{U}^*)}\boldsymbol{A}\boldsymbol{\Pi}_{\mathrm{im}(\boldsymbol{V}^*)}\rangle\\ &\geq -(1/5)\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right).\end{aligned}$$

Plugging the four inequalities that we derived back into (26), we get

$$\begin{aligned}Q &\geq (\beta/2 - \beta/20)\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle\\ &\quad + (\beta/2 - \beta/20)\langle \boldsymbol{Y}, \boldsymbol{A}^\top\boldsymbol{A}\rangle\\ &\quad - (2/5)\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right)\\ &= (9/5)\Phi(\boldsymbol{A}) - (2/5)\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right)\\ &> (3/2)\Phi(\boldsymbol{A}) - (2/5)\left(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\right).\end{aligned}$$

Finally, combining this with the smoothness inequality (23) and the lower bound on $\|\nabla g(\boldsymbol{A})\|_2^2$ (25), we derive

$$\begin{aligned}&g(\boldsymbol{A}') - g(\boldsymbol{A})\\ &\leq -(16\kappa r)^{-1}\Big(g(\boldsymbol{A}) - f(\boldsymbol{A}^*) + (1/2)\Phi(\boldsymbol{A})\Big) + 2\beta\lambda_j^2\\ &= -(16\kappa r)^{-1}\Big(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\Big) - (32\kappa r)^{-1}\Phi(\boldsymbol{A}) + 2\beta\lambda_j^2.\end{aligned}$$

What remains is the bound the sum of the last two terms. We remind the reader that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$. Now, letting $\boldsymbol{z}$ equal to the vectorized diagonal of $\boldsymbol{U}^\top\boldsymbol{W}\boldsymbol{U}$ and $\boldsymbol{\lambda}$ to the vectorized diagonal of $\boldsymbol{\Lambda}$, note that

$$\|\boldsymbol{\lambda}\|_{\boldsymbol{z}}^2 = \langle \boldsymbol{\Lambda}^2, \boldsymbol{U}^\top\boldsymbol{W}\boldsymbol{U}\rangle = \langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle,$$

using which we derive

$$\begin{aligned}\lambda_j^2 = \min_{j\in S}\lambda_j^2 &\leq \frac{\|\boldsymbol{\lambda}\|_{\boldsymbol{z}}^2}{\|\boldsymbol{z}\|_1}\\ &= \frac{\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle}{\mathrm{Tr}[\boldsymbol{U}^\top\boldsymbol{W}\boldsymbol{U}]}\\ &= \frac{\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle}{\mathrm{Tr}[\boldsymbol{U}^\top\boldsymbol{U}] - \mathrm{Tr}[\boldsymbol{U}^\top(\boldsymbol{I} - \boldsymbol{W})\boldsymbol{U}]}\\ &\leq \frac{\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle}{r' - \mathrm{Tr}[\boldsymbol{I} - \boldsymbol{W}]}\\ &\leq \frac{\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle}{r'/2}\\ &\leq \frac{\langle \boldsymbol{W}, \boldsymbol{A}\boldsymbol{A}^\top\rangle}{128r\kappa},\end{aligned}$$

where we used the fact that

$$\begin{aligned}&\mathrm{Tr}[\boldsymbol{U}^\top(\boldsymbol{I} - \boldsymbol{W})\boldsymbol{U}]\\ &= \mathrm{Tr}\left[(\boldsymbol{I} - \boldsymbol{W})^{1/2}\boldsymbol{U}\boldsymbol{U}^\top(\boldsymbol{I} - \boldsymbol{W})^{1/2}\right]\\ &\leq \mathrm{Tr}[\boldsymbol{I} - \boldsymbol{W}],\end{aligned}$$

because the columns of $\boldsymbol{U}$ are orthonormal. We also used the property that $\mathrm{Tr}[\boldsymbol{I} - \boldsymbol{W}] \leq r'/2$ and the fact that $r' \geq 256r\kappa$ by the lemma statement.

Similarly, we derive that

$$\lambda_j^2 \leq \frac{\langle \boldsymbol{Y}, \boldsymbol{A}^\top\boldsymbol{A}\rangle}{128r\kappa},$$

and, adding these two inequalities, we have

$$2\beta\lambda_j^2 \leq (32r\kappa)^{-1}\Phi(\boldsymbol{A}),$$

finally concluding that

$$g(\boldsymbol{A}') - g(\boldsymbol{A}) \leq -(16\kappa r)^{-1}\Big(g(\boldsymbol{A}) - f(\boldsymbol{A}^*)\Big).$$

□

E. Lower Bounds

Lemma E.1 (IHT lower bound)**.** *Let* $f(\boldsymbol{x}) := (1/2)\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$. *For any* $\kappa, s \geq 1$, $s' \leq 0.6s\kappa^2$, *there exists a (diagonal) matrix* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *and a vector* $\boldsymbol{b} \in \mathbb{R}^n$ *where* $n = s(\kappa^2 + \kappa + 1)$, f *is* 1*-strongly convex and* κ*-smooth, as well as an* s*-sparse solution* $\boldsymbol{x}^*$ *and an* s'*-sparse solution* $\boldsymbol{x}$, *such that*

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*) + 0.1s\kappa^2$$

but

$$\boldsymbol{x} = H_{s'}\left(\boldsymbol{x} - \beta^{-1}\nabla f(\boldsymbol{x})\right),$$

i.e. $\boldsymbol{x}$ *is a fixpoint for IHT.*

Proof. We use the same example as in (Axiotis & Sviridenko, 2021b), Section 5.2: $\boldsymbol{A}$ is diagonal with

$$\boldsymbol{A}_{ii} = \begin{cases}1 & \text{if } i \in I_1\\ \sqrt{\kappa} & \text{if } i \in I_2\\ 1 & \text{if } i \in I_3,\end{cases}$$

where $I_1 = [s], I_2 = [s+1, s(\kappa+1)], I_3 = [s(\kappa+1)+1, s(\kappa^2+\kappa+1)]$, and $\boldsymbol{b}$ is defined as

$$b_i = \begin{cases}\kappa\sqrt{1-4\delta} & \text{if } i \in I_1\\ \sqrt{\kappa}\sqrt{1-2\delta} & \text{if } i \in I_2\\ 1 & \text{if } i \in I_3,\end{cases}$$

for some sufficiently small $\delta > 0$ used for tie-breaking. We define

$$x_i^* = \begin{cases}\kappa\sqrt{1-4\delta} & \text{if } i \in I_1\\ 0 & \text{otherwise}\end{cases}$$

and, for some arbitrary s'-sized $S \subseteq I_3$

$$x_i = \begin{cases} 0 & \text{if } i \in I_1 \cup I_2 \cup I_3 \backslash S \\ 1 & \text{otherwise}\,. \end{cases}$$

Note that $f(\boldsymbol{x}) - f(\boldsymbol{x}^*) = 0.5s\kappa^2(1-4\delta) - 0.5s' \geq 0.1s\kappa^2$.
Furthermore, the gradient is equal to

$$\begin{aligned} \nabla f(\boldsymbol{x}) &= \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\ &= \begin{cases} -\kappa\sqrt{1-4\delta} & \text{if } i \in I_1 \\ -\kappa\sqrt{1-2\delta} & \text{if } i \in I_2 \\ -1 & \text{if } i \in I_3 \backslash S \\ 0 & \text{if } i \in S\,, \end{cases} \end{aligned}$$

and since we have $\beta = \kappa$,

$$\boldsymbol{x} - \beta^{-1}\nabla f(\boldsymbol{x}) = \begin{cases} \sqrt{1-4\delta} & \text{if } i \in I_1 \\ \sqrt{1-2\delta} & \text{if } i \in I_2 \\ 1/\kappa & \text{if } i \in I_3 \backslash S \\ 1 & \text{if } i \in S\,, \end{cases}$$

implying that $H_{s'}\left(\boldsymbol{x} - \beta^{-1}\nabla f(\boldsymbol{x})\right) = \boldsymbol{x}$. □